國立臺灣大學管理學院資訊管理學系

碩士論文

Department of Information Management

College of Management

National Taiwan University

Master Thesis

運用最佳化技術於綠能無線感測網路能耗控制機制

An Optimization-based Power Control Mechanism for Saving Energy in Green Wireless Sensor Networks

劉柏辰

BO-CHEN LIU

指導教授: 林永松 博士

Advisor: Frank Yeong-Sung Lin, Ph.D.

中華民國 112 年 1 月

January, 2023

doi:10.6342/NTU202210101

摘要

近年來，無線感測網路因為其成本低，耗電量低，體積小和容易布置等等特性，被大量的運用在各領域中。但其缺點也顯而易見。因為每個感測器中是以無線的方式傳送資料，因此電源的來源通常來自感測器內的電池。感測網路中能源的節省和功率控制便成為一個極其重要的議題。

在此論文中我們將此無線網路最小化之問題構建為數學模型，並且此問題受延遲和產出之限制。在模型中我們需要決定感測器活躍的機率，傳送之距離和傳送之封包大小並藉以最小化能耗。此些決策變數中存在著不同的平衡，我們也藉由後續的實驗找出他們之間不同的關係。

此論文於運用拉格朗日鬆弛法，將其分解成子問題一一解出並最終產出緊貼於上界之下界數值。在實驗中我們也找出不同拉格朗日係數的使用方法，並期望能達到能源消耗的最小化，並同時也能維持網路的連接和資料之輸出。

關鍵字：功率控制，能耗節省，綠能科技，無線感測網路，拉格朗日鬆弛法，最佳化

doi:10.6342/NTU202210101

Abstract

In recent years, Wireless Sensor Networks(WSNs) have been widely used in various fields because of their low cost, low power consumption, small size and easy deployment. But its shortcomings are also obvious. Because each sensor transmits data wirelessly, the source of power usually comes from the battery inside the sensor. Energy conservation and power control in sensing networks becomes an extremely important issue.

In this paper we model this wireless network minimization problem as a mathematical model, and the problem subjected to delay and throughput constraints. In the model we need to determine the probability of the sensor being active, the distance to transmit and the size of the transmitted packet to minimize energy consumption. There are different trade offs among these decision variables, and we also find different relationships between them through follow-up experiments.

This paper uses the Lagrangian relaxation method and decompose it into sub-problems and solve them one by one, and find a lower bound that is tight with the upper bound value. In the experiment, we also find out different use of the Lagrangian multipliers, and hope to achieve the minimization of energy consumption, and at the same time to maintain the network connection and data output.

doi:10.6342/NTU202210101

Keywords: Power Control, Energy-Saving, Green Information Technology, Wireless Sensor Network, Lagrangian Relaxation, Optimization

doi:10.6342/NTU202210101

Contents

doi:10.6342/NTU202210101

doi:10.6342/NTU202210101

doi:10.6342/NTU202210101

doi:10.6342/NTU202210101

doi:10.6342/NTU202210101

List of Figures

doi:10.6342/NTU202210101

List of Tables

doi:10.6342/NTU202210101

doi:10.6342/NTU202210101

doi:10.6342/NTU202210101

Chapter 1 Introduction

1.1 Background Overview

1.1.1 Characteristics of WSN

Wireless sensor network (WSN) consists of a large number of sensing nodes which are responsible for monitoring or performing certain measurements, such as temperature, humidity, vehicular movement, noise levels, pressure, soil makeup [1],etc.

Nodes in sensor networks are also mostly battery powered [2]. Because of this characteristic, designing the networks with high energy efficiency becomes crucial in order to maximize their lifetime [3]. Properties such as decentralized control, broadcast and channel to transmit , among others are derived from ad-hoc networks. However, WSN are still different from traditional ad-hoc networks.

The way that ad-hoc networks communicates is any-to any because the nodes in ad-hoc network are in general less energy constrained [4]. On the other hand,the paradigm

doi:10.6342/NTU202210101

of many-to-one communication is more common in sensor networks [5]. This is because the collected data will be sent to the nearest base station for aggregating the resulting data to a higher class for processing [6], which later forms different kinds of topology and will be changed frequently [7].

1.1.2 Architecture of WSN

Figure 1.1 demonstrates the general architecture of a WSN. It can be observed from the diagram that sensor nodes are randomly deployed which formed a coverage area. We suppose that sensor nodes are all capable of collecting and sensing information in their area of interest. Each nodes are also able to communicate with higher class nodes for transmitting the collected data through network interfaces. Higher class nodes will be determined from the range with the sink/base station, which are used to gather data from lower class nodes. A sink/base station is a node that acts as a bond between the user's internet and the WSN. Network data are often processed by the sink/base station in preparation for sending only relevant data to the user. Requests to other nodes can also be sent by the user via the sink/base station [8].

doi:10.6342/NTU202210101

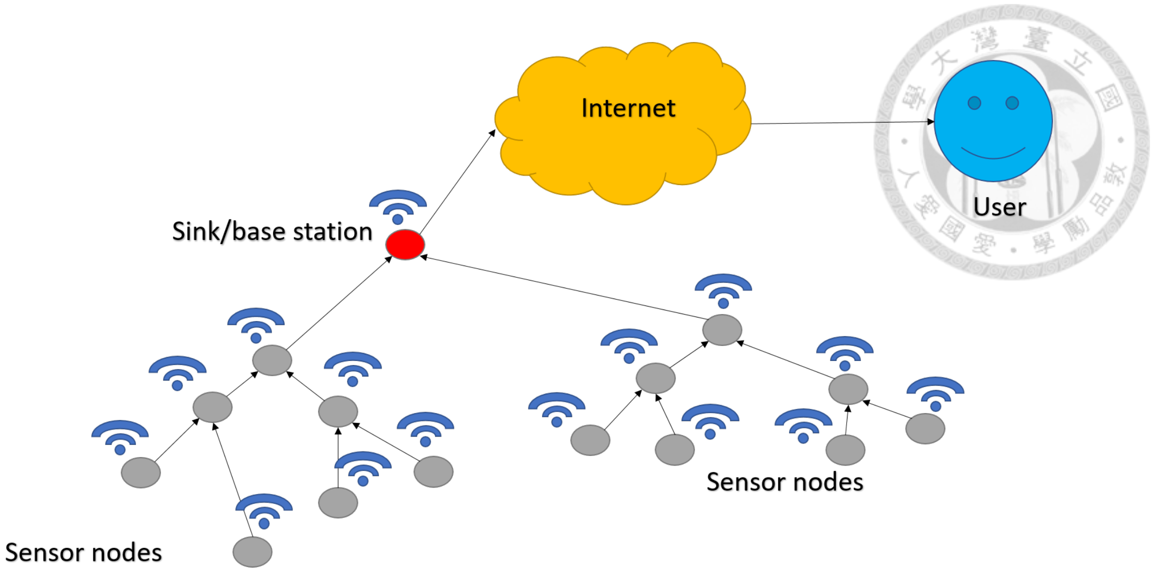

Figure 1.1: Architecture of WSN

It is noticed that high level nodes closer to the base station or the sink are more easily to exhaust their battery energy than other lower level nodes since they are in charge of aggregating data [9]. Hence,sensor nodes should optimize several decisions ,including the amount of data flow, transmission power level and the activeness of the sensor on each link in order to avoid the over-utilization of energy in batteries [10].

1.2 Motivation

WSN plays an important role in the development for a wide range of application in Internet of Things(IoT) [11]. It is also considered one of the emerging technologies that will change the world [12], [13]. However, due to environmental constraints and other factors that increases costs, changing the batteries of the sensor seems to be difficult. For instance, it is not cost-effective and even impossible to replace the batteries of the nodes in

doi:10.6342/NTU202210101

a WSN deployed underneath the ocean for observant purpose. Hence, energy efficiency is one of the most crucial criteria while designing WSN [14] and has not stop being discussed even in recent years [15]. In this research, we intend to propose a model to optimize the energy efficiency of a WSN.

1.3 Objectives

Lifetime maximization of WSN has been addressed by substantial research works [16], [17], [18], [19]. Energy conservation strategies are generally divided into several types [20], [21]:

- Type 1 : Scheduling strategy on alternating states of nodes between sleep and active modes.
- Type 2 : Power control aiming to optimize tradeoff between energy consumption and connectivity.
- Type 3 : Best routes,cluster and aggregation for sensor nodes.
- Type 4 : Reduction or compression of transmitted data .
- Type 5 : Efficiency of retransmission and acknowledgements protocols on data link layer.

doi:10.6342/NTU202210101

Most studies concern their power control mechanism on a certain type. It is shown that [22] focuses their study on type 2, and [21] on the other hand extend their research on type 1. [23] took both transmission power and packet size into consideration. When the packet size is reduced, the overall impact of bit error rates on packet loss will also drop. However, a smaller packet size will result in more packet transmission due to the fixed header protocol. On the other hand, by increasing the transmission power of sensor nodes, packet loss probability will decrease, but high transmission power will also result in high energy dissipation. The tradoff between packet size and transmission power are discussed specifically in [23]. Several researches also simplify assumptions such as assuming perfect feedback channel and lossless channel [24], [25], [26]. However, in practical, it is known that WSNs are subjected to packet errors so as acknowledgements sent by receivers.

1.3.1 Methods

In this research, we tend to include the studies on type 1 and 2. Therefore, in order to address the issues of the states between nodes, the tradeoff between energy consumption and connectivity will be discuss in our model. The efficiency of retransmission and acknowledgements protocols on data link layer will also be considered. A mathematical formulation taking both packet size and transmission power into account as well as different solution approaches will be introduced in the following chapters.

doi:10.6342/NTU202210101

In the first part of our proposed model, we focus on formulating the relationship between two sensor nodes, which forms an one-to-one link. In the second part, an many-to-one relationship is considered, which will later form a star topology and an aggregate point for gathering data from the lower class. At last, a tree structure will be discussed in order to form a complete WSN.

1.4 Thesis Organization

This thesis is organized as follows. In Chapter 2, related works regarding this problem will be provided. A formal formulation of the problem, including text description as well as in mathematical form, will be described in Chapter 3. Chapter 4 will provide the solution approach for the problem stated in Chapter 3. Experimental results and discussion on the previously proposed methods will be shown in Chapter 5 . Finally, a conclusion will be drawn in Chapter 6.

doi:10.6342/NTU202210101

Chapter 2 Literature Review

Mechanisms to reduce energy consumption in wireless sensor network has been studied extensively. By determining the states of nodes through its active and sleeping periods brightly can help us avoid needless waste of energy. Power control techniques that seek to optimal the tradeoff of energy consumption and connectivity are also crucial. Last but not least, finding the best routes, cluster and aggregation points when designing networks will also prevent energy waste. In this section, studies and technology related to these research will be presented.

2.1 Duty Cycle

Duty cycle is one of the most effective operation in terms of increasing the energy efficiency of the sensor network. It managed the energy resource of the nodes by constantly switching the states of the sensor nodes. There are different operation states of a node, such as idle,sleep,listen and transmit. Distance-based Duty Cycle Assignment (DDCA)

doi:10.6342/NTU202210101

and Traffic-Adaptive Distance-based Duty Cycle Assignment (TDDCA) are proposed in [27]. Here it assumes that nodes closer to sink must transmit a larger size of packet than those further from the sink. Hence , DDCA is used as a function to determine the duty cycle with the distances of the nodes to the sink. Additionally, TDDCA is responsible for adapting duty cycle later on according to the current traffic patterns observed by the nodes. According to receiver-based protocols [28], the traffic of the network can be indicated by the number of retransmitted RTS packets. If the retransmission of RTS increases dramatically and eventually outnumbers the original RTS packet, it means that there might be traffic congestion. TDDCA will then tune the duty cycle to mitigate congestion.

Coverage requirements are also related to the node duty cycle. Not only should we minimize the consumption of energy, but the sensor nodes should also be able to cover the targets that is monitored. Moreover,a disjoint set of nodes should be formed to cover the monitored area in order to maximize the network's lifetime. However,connectivity of the network should also be considered. [29] focus on solving k-coverage problem while ensuring the connectivity between all active nodes. K-coverage means that every location is covered by at least k sensors in a sensor network. By solving this, [29] decomposes the problem into field slicing and sensor selection. Field slicing first split the sensing region into little pieces of a particular shape(Reuleaux Triangles), then sensor selection selects as small subset of sensors as possible to cover it.

doi:10.6342/NTU202210101

2.2 Control of Transmission Power

The control of transmission power aims to adjust the node's transmission power at appropriate levels due to factors such as the range between transmitter and receiver or the current state of traffic. Researches about this topic has been conducted in [30], [31].It is shown that [31] classifies different approaches of power control the protocol layers employed: MAC, Network and Transport layer. A MAC layer approach aims to decrease the chance of collision,in order to minimize the energy consumption used in transmission. Network layer approach employed two basic scenes: Power-Aware Routing and Maximum Lifetime Routing. Both of the approach concentrates on network routing rather than control in transmission power, which we will discuss in the next section.Transport layer Protocol (TCP) is responsible for congestion control and retransmission in a network,so approaches employed in transport layer can alter the retransmission behavior in order to achieve lower energy consumption [32]. Power control in transmission problem are also referred to as the Range Assignment(RA) or Strong Minimum Energy Topology(SMET), which discusses the tradeoffs of throughput,traffic and reliability.

In [30], an Adaptive Transmission Power Control(ATPC) model was proposed. In ATPC, a model built in each nodes will adjust their power according to the link quality. This model employs a feedback-based transmission power control algorithm to maintain

doi:10.6342/NTU202210101

link quality dynamically. ATPC relies on pairwise adjustment to achieve energy saving and furthermore shows the superiority of link level compared to the node level and network level in terms of energy efficiency [33].It is shown that [33] built a mathematical model to investigate the impact of transmission power control to network lifetime. Higher transmission power will sure decrease the probability of handshake failure, but it may not lead to maximizing the network's lifetime since some links selection of maximum might increase energy consumption. Lossless feedback channel are also investigated in [33]. [22] proposed a hybrid model by considering both transmission power and the node's address. The transmission power will be optimized according to the distance between neighbor nodes and through searching the table with the address of nodes for the next hop. Therefore, transmission can be adjust with the information of the mapping table.

2.3 Topology Control and Routing

A decent node deployment in topology control can lead to reducing network traffic, avoid packet collision , improving network throughput and save energy. The main part of node deployment is to find a subset of nodes that is strongly connected to become the backbone of the network. The rest of the nodes can be connected to the backbone. This backbone topology not only guarantees the connectivity of the network but also allows non-backbone nodes to be turned off to save energy. This kind of problem is often modeled

doi:10.6342/NTU202210101

as Connected Dominating Set (CDS) mathematical problem. A CDS-based algorithm is presented by [34] to construct the network with prolonging the network lifetime and balancing energy consumption. But when it comes to fault tolerance, CDS problems won't be enough because it only preserves 1-connectivity. Therefore, kmCDS problem where k-connectivity and m dominating sets are brought to consideration.

Routing, also known as the data transmission problem, is broadly studied in WSN. It can be roughly divided into the group related to Shortest Path Tree (SPT) and Minimum Spanning Tree (MST) based models and those centered around flow problems. SPT and MST algorithms help us find the paths that consumes the minimum energy consumption to achieve energy efficiency [35], [36]. Dijkstra's or Bellman Ford algorithms are often used in these models. Different situation of the nodes are took into consideration by [37] such as some nodes may deplete energy faster than others. However, SPT may lead to unbalance load between sensors since the model tends to choose certain routes, so residual energy, buffer size or other factors are later took into account in [38]. Routing problems aim to minimize total energy consumption or maximizing networks lifetime are also formulate as Multi-Commodity Flow Problems. Commodity is a source-destination pair, and it is shown in [39]that the multi-commodity flow problem is NP-hard. [40] formulates the Multi-commodity flow problem into Integer Linear Programming and represents the flow by the number of packets and the transmission energy. [41] proposed an energy-aware

doi:10.6342/NTU202210101

routing algorithm to prolong the network life time of wireless sensor network.

doi:10.6342/NTU202210101

Chapter 3 Problem Formulation

In this chapter, the optimization problem of power control mechanism for saving energy in wireless sensor networks will be thoroughly described. The constraints as well as the associated assumptions will also be covered. A mathematical formulation will be presented at the end of this chapter.

3.1 Problem Description

The aim of this research is to minimize the energy consumption of the WSN while considering the connectivity and throughput. We proposed three models as a slotted time system which discusses the optimal problem based on three perspective of the nodes: One-to-One Relationship, Many-to-One Relationship and the whole Network Tree Structure Relationship. In a slotted time system, collision will occur if more than 2 node pairs are competing for a same slot. The bandwidth of each time slot is considered fixed. We will discuss these three models sequentially. An overview of each model will be present in the

doi:10.6342/NTU202210101

following section.

3.2 Model 1 : One-to-One Relationship

In this section, the mathematical form is given for the problem of One-to-One Relationship model where we take node i and node j into consideration.

The given parameters as well as their descriptions are shown in Table 3.1:

Table 3.1: Given Parameters

Notation	Description
S	The index set of sensors, which is $\{1, 2, 3, \ldots, \overline{s}\}$
L	The index set of all possible links, which is $\{1, 2, 3, \ldots, \overline{l}\}$
$\overline{T_{ij}}$	The allowable delay from $i \in S$ to $j \in S$ (end to end)
λ_{ij}	Data rate between $i \in S$ and $j \in S$
d_{ij}	The distance between $i \in S$ and $j \in S$
R_i	Set of possible range for $i \in S$
τ_i	Timeout interval for $i \in S$ (a given $\sharp$ of time slots)
t_{ij}	The transmission time from $i \in S$ to $j \in S$, e.g. slot time = 1
g_{ij}	The aggregate flow on link ($i \in S$,$j \in S$)$\in L(\frac{\lambda_{ij}}{q_i q_j P_{ij}})$
f_{ij}	Retransmission

doi:10.6342/NTU202210101

The decision variables and their descriptions are are shown in Table 3.2:

Table 3.2: Decision Variables

Notation	Description
r_s	Transmission range of $s \in S$,$r_s \in R_s$
M	The index set of all possible packet size , which is $\{1, 2, 3, \ldots, \overline{m}\}$
q_i	The probability that $i \in S$ is active in a time slot
$P_{ij}(r_i, m)$	The probability of $i \in S$ to transmit packet with $m \in M$ size to $j \in S$ when no error occurs with transmission range radius of $r_i \in R_i$
	Also the function of j, which is the basic inherent channel quality between i and j
$C_i^a(r_i, m)$	The average power consumption rate when $i \in S$ is active with transmission range r_i in 1 time slot(influenced by m)
$C_i^b(r_i, m)$	The average power consumption rate when $i \in S$ is inactive with transmission range r_i in 1 time slot(influenced by m)
$C_i^\tau(r_i, m)$	The average power consumption rate for $i \in S$ to transmit a $m \in M$ size packet(in 1 time slot) with transmission range $r_i \in R_i$
	Also the functions $C_i^a(r_i, m)$,$C_i^b(r_i, m)$ and $C_i^\tau(r_i, m)$ are given, the parameters of the function are to be determined, so the value of the function itself is a decision variable

The objective function is expressed as formula (3.1):

$$\min \quad \sum_{i\in S}\sum_{j\in S} \frac{\left[C_i^a(r_i,m)q_i + C_j^a(r_j,m)q_j + C_i^b(r_i,m)(1-q_i) + C_j^b(r_j,m)(1-q_j) + C_i^\tau(r_i,m)q_i\right]}{m} \tag{3.1}$$

The objective function aims to minimize the energy consumption in a time slot by controlling q_i and q_j, which is the probability that node i and node j is active in a time slot.

In $C_i^a(r_i, m)$ we consider the transmission range of node i to decide the power consumption when the node is active and in $C_j^b(r_i, m)$ we consider the transmission range of node i to decide the power consumption when the node is inactive. When transmitting data,

doi:10.6342/NTU202210101

we consider the transmission range to determine the power consumption of $C_i^\tau(r_i, m)$. We also consider the packet size level that node i is transmitting because it would influence the size of the time slot. Therefore, the bigger the packet size, the longer the time slot will be needed to transmit a packet. The probability of transmitting a packet without error will also decrease when m increases, and we assume each packet has a fixed size header, therefore the larger the packet size is, the larger the throughput is. We take all the above mentioned factors into consideration to find the trade off of the packet size and the power consumption in a single byte. So the objective function will be divided by m to normalize by the length of time slot.

The packet size is categorized into different level where high packet size level indicates larger packet size. At a same encapsulation mechanism, packets share the same overhead. So a larger packet size indicates a larger payload size, which means more data will be transmitted. On the other hand, with a fixed bit error rate, the larger of the packet size level, the larger of the power consumption $C_i^\tau(r_i, m)$. As for the success probability($P_{ij}(r_i, m)$) of node i to transmit packet size level m to node j with transmission range radius r_i , the larger of the packet size level, the smaller of $P_{ij}(r_i, m)$. Hence there appears an interesting tradeoff, which is when in a noiseless channel, larger packet size may lead to larger throughput. But in a noisy channel, larger packet size may cause lower probability to success.

doi:10.6342/NTU202210101

Figure 3.1 shows the relationship between node i and node j, with the success and error rate to transmit a packet of $q_i q_j P_{ij}(r_i, m)$ and $1 - q_i q_j P_{ij}(r_i, m)$ respectively. $\lambda_{ij} + f_{ij}$ is the aggregate flow on link i, j. Each transmission is considered as a bernoulli trial as result of the same probability to successfully transmit. We assume that we have to transmit k times to success, each failure transmission will cost us a timeout interval of τ_i. Hence, the transmission time is a random variable governed by a geometric distribution. The expected value for the number of transmission to get the first success is a fraction of the probability to successfully transmit, which is $\frac{1}{q_i q_j P_{ij}(r_i,m)}$. The probability $P_{ij}(r_i, m)$ considers when both data and acknowledgement success. If the acknowledgement is received by the transmitter before a timeout interval τ_i, then it is considered a successful transmission.

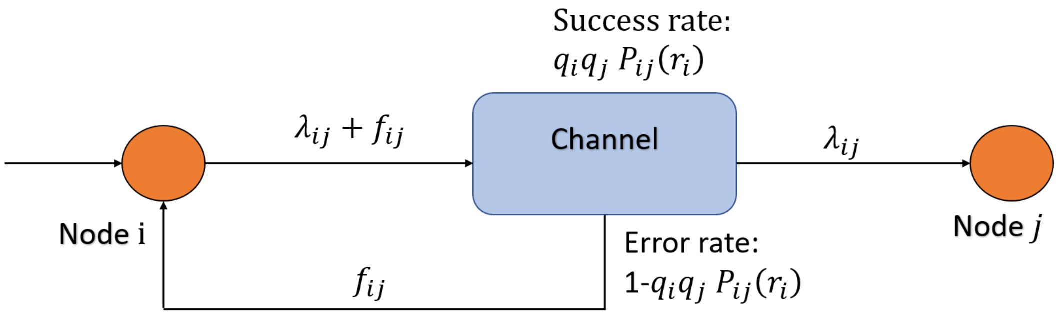

Figure 3.1: One-to-One Relationship

If the packet successfully transmit to node j, data rate of λ_{ij} will be transferred and formula (3.2) will be derived as follow:

$$\lambda_{ij} = (\lambda_{ij} + f_{ij})(q_i q_j P_{ij}(r_i, m)) \qquad \forall i \in S, j \in S, m \in M \tag{3.2}$$

When the transmission failed, a retransmission will begin and f_{ij} can be derived as follow

doi:10.6342/NTU202210101

formula (3.3):

$$f_{ij} = (\lambda_{ij} + f_{ij})(1 - q_i q_j P_{ij}(r_i, m)) \qquad \forall i \in S, j \in S, m \in M \tag{3.3}$$

The value of f_{ij} can also be derived from formula (3.2) as formula (3.4):

$$f_{ij} = \frac{\lambda_{ij}(1 - q_i q_j P_{ij}(r_i, m))}{q_i q_j P_{ij}(r_i, m)} \qquad \forall i \in S, j \in S, m \in M \tag{3.4}$$

The objective function is subject to the following constraints.

There are several ways to recover from error transmission which occurs retransmission delay. The first one we introduce is error detection. We assume that a successful transmission includes the acknowledgement send from the receiver which shows that the transmission is without error, otherwise the receiver will send a negative acknowledgement. The second one is when the sender did not receive acknowledgement, there will be a timeout interval denoted as τ_i to keep the sender from infinite waiting for the acknowledgement.

Constraint (3.5) ensures that the time spent for a single successful transmission will be smaller than the allowable delay from node i to node j. We assume that queuing delay is ignored, the delay over link (i,j) is the time spent for a single successful transmission, which is the number of transmission before getting the first success $(\frac{1}{q_i q_j P_{ij}(r_i,m)} - 1)$ times the timeout interval τ_i emerged for each failed transmission plus one slot time for the success transmission. It is expressed as

$$\tau_i \left[\frac{1}{q_i q_j P_{ij}(r_i, m)} - 1 \right] + 1 \leq \overline{T_{ij}} \qquad \forall i \in S, j \in S, m \in M \tag{3.5}$$

doi:10.6342/NTU202210101

τ_i is the smaller the better in a single slot. However there are other delays so we have to precisely set up τ_i.

Constraint (3.6), (3.7) and (3.8) ensures that the value of q_i , q_j and $P_{ij}(r_i, m)$ fall within a small number ϵ and 1. Since q_i, q_j and $P_{ij}(r_i, m)$ denotes the probability that node i is active in a time slot, the probability that node j is active in a time slot and the probability of node i to transmit packet to node j without error with transmission range radius of r_i, respectively, it is required that q_i , q_j and $P_{ij}(r_i, m)$ fall in between a small number ϵ and 1. The constraints are expressed as :

$$\epsilon \leq q_i \leq 1 \qquad \forall i \in S \tag{3.6}$$

$$\epsilon \leq q_j \leq 1 \qquad \forall j \in S \tag{3.7}$$

$$\epsilon \leq P_{ij}(r_i, m) \leq 1 \qquad \forall i \in S, j \in S \tag{3.8}$$

Constraint (3.9) and (3.10) are expressed as

$$r_i \in R_i \qquad \forall i \in S \tag{3.9}$$

$$r_j \in R_j \qquad \forall j \in S \tag{3.10}$$

doi:10.6342/NTU202210101

3.3 Model 2 : Many-to-One Relationship

The given parameters as well as their descriptions are shown in Table 3.3:

Table 3.3: Given Parameters

Notation	Description
S	The index set of sensor nodes, which is $\{1, 2, 3, \ldots, \overline{s}\}$
L	The index set of all possible links, which is $\{1, 2, 3, \ldots, \overline{l}\}$
κ	The relay node , responsible for aggregating data sent from the lower level layer sensor nodes
$\overline{T_{i\kappa}}$	The allowable end to end delay from $i \in S$ to relay node κ
n	The amount of nodes in the lower layer of Many-to-one structure
$d_{i\kappa}$	The distance between $i \in S$ and relay node κ
R_i	Set of possible range for $i \in S$
R_κ	Set of possible range for relay node κ
τ_i	Timeout interval for $i \in S$ (a given $\sharp$ of time slots)
$t_{i\kappa}$	The transmission time from $i \in S$ to relay node κ , e.g. 1 slot time
$g_{i\kappa}$	The aggregate flow on link ($i \in S$,relay node κ)$\in L(\frac{1}{q_i q_\kappa P_{i\kappa}})$
t	The least expected time slots for the network to work
P_i	The initial power storage for sensor node $i \in S$
P_κ	The initial power storage for relay node κ

doi:10.6342/NTU202210101

The decision variables and their descriptions are are shown in Table 3.4:

Table 3.4: Decision Variables

Notation	Description
r_s	Transmission range of $s \in S$,$r_s \in R_s$
r_κ	Transmission range of relay node κ, $r_\kappa \in R_\kappa$
M	The index set of all possible packet size , which is $\{1, 2, 3, \ldots, \overline{m}\}$
q_i	The probability that $i \in S$ is active in a time slot
q_κ	The probability that relay node κ is active in a time slot
$P_{i\kappa}(r_i, m)$	The probability of $i \in S$ to transmit packet with $m \in M$ size to relay node κ when no error occurs with transmission range radius of $r_i \in R_i$
$C_i^a(r_i, m)$	The average power consumption rate when $i \in S$ is active with transmission range $r_i \in R_i$ in 1 time slot(influenced by m)
$C_i^b(r_i, m)$	The average power consumption rate when $i \in S$ is inactive with transmission range $r_i \in R_i$ in 1 time slot(influenced by m)
$C_\kappa^a(r_\kappa, m)$	The average power consumption rate when relay node κ is active with transmission range $r_\kappa \in R_\kappa$ in 1 time slot(influenced by m)
$C_\kappa^b(r_\kappa, m)$	The average power consumption rate when relay node κ is inactive with transmission range $r_\kappa \in R_\kappa$ in 1 time slot(influenced by m)
$C_i^\tau(r_i, m)$	The average power consumption rate for $i \in S$ to transmit a $m \in M$ size packet(in 1 time slot) with transmission range $r_i \in R_i$
	Also the functions $C_i^a(r_i, m)$, $C_i^b(r_i, m)$, $C_\kappa^a(r_\kappa, m)$, $C_\kappa^b(r_\kappa, m)$ and $C_i^\tau(r_i, m)$ are given, the parameters of the function are to be determined, so the value of the function itself is a decision variable

The Many-to-One Relationship model is an extension from the One-to-One Relationship model. As shown in Figure 3.2, we assume that there are n nodes waiting to transmit data to relay node κ and they share the same interface to transmit. N nodes indicates that each node's probability to transmit will not exceed $\frac{1}{n}$, which constrained to the maximum throughput. Moreover, the probability to transmit of each nodes might be less than $\frac{1}{n}$ in order to save energy. Relay node κ is the aggregate node that is responsible for gathering

doi:10.6342/NTU202210101

data from the lower level nodes, and so theoretically relay node κ will have a higher probability to be active, which is a higher q_κ. Therefore, we can consider this Many-to-One Relationship model as a tree structure.

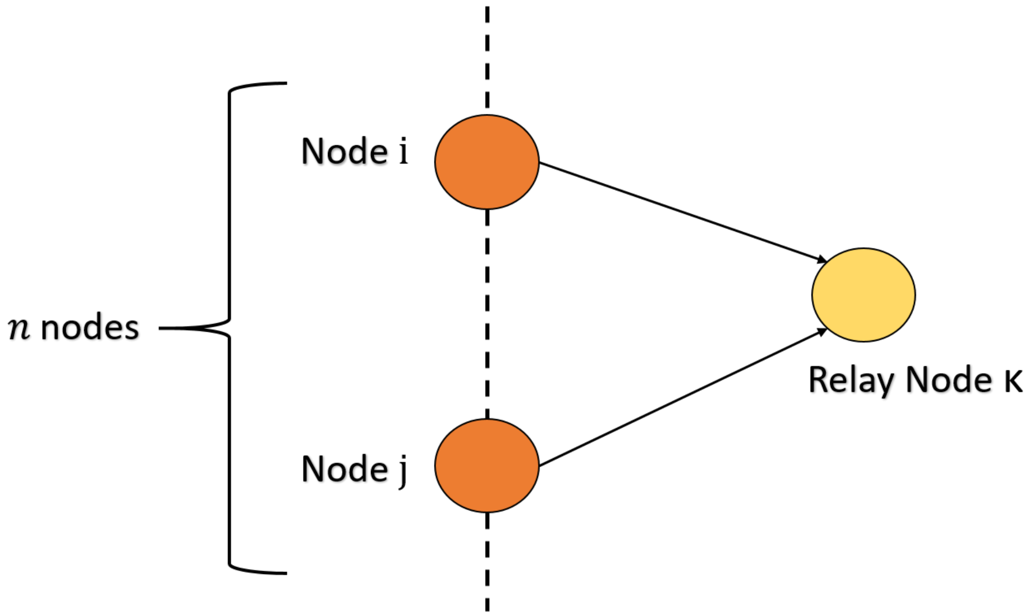

Figure 3.2: Many-to-One Relationship

In this tree structure, we first take only node i and node j into consideration. We assume that node i and node j is in need to transmit data to relay node κ , so node i and node j is competing for relay node κ . We can assume that node i and node j obtains different average delay upper limit due to the importance of the data. The probability for i to successfully transmit to k is when node i and relay node κ is active, but node j is inactive in order to avoid collision. And we assume that lower level sensor nodes will always have data to transmit to relay node κ. The formula indicating the possibility of sensor node i to successfully transmit to relay node κ is shown below as formula (3.11).

$$q_i(1-q_j)q_\kappa P_{i\kappa}(r_i, m) \qquad \forall i \in S, j \in S, m \in M \tag{3.11}$$

doi:10.6342/NTU202210101

When all the nodes finish transferring their data to relay node κ , then it is consider that relay node κ has finish gathering all the data. In this scenario that we only consider node i and node j, the time that relay node κ will take to finish aggregating all the data is the largest value between the time node i transmits and the time node j transmits. It is shown as formula (3.12).

$$\max\{\tau_i \left[\frac{1}{q_i(1-q_j)q_\kappa P_{i\kappa}(r_i, M)} - 1\right] + 1, \tau_i \left[\frac{1}{q_j(1-q_i)q_\kappa P_{jk}(r_j, M)} - 1\right] + 1\}$$

$$\forall i \in S, j \in S, m \in M \tag{3.12}$$

Now we take all the n nodes in the lower layer into consideration, which is node a to node z competing to transmit to relay node κ . In this scenario we assume that there are always data needed to be send in the lower layer, which is when node a to node z is active, they will try to send data to relay node κ . We focus on the possibility where node i successfully transmits data to relay node κ . This system we designed is a slotted time system, which means for every time slot, if there is only one node transmitting data then it will success, else it will collide and fail to transmit. We also neglect the fact that nodes are waiting for timeout interval to timeout or the situation that the previous time sending the data is failure, which indicates that the probability of no one is using a time slot is low. Therefore for i to successfully transmit, it happens at the scenario where all the nodes are inactive except the transmitter node i and the receiver relay node κ. In the scenario we consider, we assume that each point is parallel and competing for an interface in a slotted

doi:10.6342/NTU202210101

time. Every sensor has information to send and will send when there is a chance. The probability for node i to successfully transmit to relay node κ is shown as formula (3.13).

$$q_i(1-q_a)(1-q_j)\times ... \times(1-q_z)q_\kappa P_{i\kappa}(r_i,M) \tag{3.13}$$

The objective function is expressed as formula (3.14):

$$\begin{aligned}\min \quad & \frac{C_i^a(r_i,m)q_i + C_j^a(r_i,m)q_j + C_a^a(r_a,m)q_a + ... + C_z^a(r_z,m)q_z + C_\kappa^a(r_\kappa,m)q_\kappa}{m} \\ & + \frac{C_i^b(r_i,m)(1-q_i) + C_j^b(r_i,m)(1-q_j) + ... + C_z^b(r_z,m)(1-q_z) + C_\kappa^b(r_\kappa,m)(1-q_\kappa)}{m} \\ & + \frac{C_i^\tau(r_i,m)q_i}{m} + \frac{C_j^\tau(r_j,m)q_j}{m} + \frac{C_a^\tau(r_a,m)q_a}{m} + ... + \frac{C_z^\tau(r_z,m)q_z}{m}\end{aligned} \tag{3.14}$$

The objective function (3.14) aims to minimize the energy consumption by controlling q_a to q_z,which is the probability that node a to node z is active in a time slot. The first row of the formula indicates the energy consumption where node a to node z is active. The second row shows the energy consumption where node a to node z is inactive, and the last row shows the energy consumption where nodes are transmitting data to relay node κ. The energy consumption where nodes are transmitting data to relay node κ is the probability when nodes are active times the the average power consumption rate. The reason to multiply the probability to transmit to the energy consumption where nodes are transmitting data is because the nodes won't always send data in every time slot, it also depends on the probability of activeness. We also consider the packet size level that node i is transmitting because it would influence the size of the time slot. Therefore, the bigger the packet size,

doi:10.6342/NTU202210101

the longer the time slot will be needed to transmit a packet. The probability of transmitting a packet without error will also decrease when m increases, and we assume each packet has a fixed size header, therefore the larger the packet size is, the larger the throughput is. We take all the above mentioned factors into consideration to find the trade off of the packet size and the power consumption in a single byte. So the objective function will be divided by m to normalize by the length of time slot.

Based on fairness, we can also consider formula (3.15):

$$q_i = q_j \qquad \forall i \in S, j \in S \tag{3.15}$$

Each q_i can be different, and formula wise is absolutely feasible. But when solving the problem, it will become very difficult to solve. After expansion, you will find that there will be various combinations of terms multiplied together. Although it is doable, the complexity of the problem will also increase rapidly. Therefore we consider formula (3.15).

However for q_κ, relay node κ is the aggregate node that is responsible for gathering data from the lower level nodes, and so theoretically relay node κ will have a higher probability to be active, which results in a higher q_κ.

The new success rate for node i to transmit is shown as formula (3.16):

$$q_i(1-q_i)^{(n-1)} q_\kappa P_{i\kappa}(r_i, m) \qquad \forall i \in S, m \in M \tag{3.16}$$

The new objective function based on formula (3.15) is shown as formula (3.17):

doi:10.6342/NTU202210101

$$\min \quad \sum_{i \in S} \left[\frac{C_i^a(r_i, m)q_i + C_\kappa^a(r_\kappa, m)q_\kappa + C_i^b(r_i, m)(1-q_i) + C_k^b(r_\kappa, m)(1-q_\kappa) + C_i^\tau(r_i, m)q_i}{m} \right] \tag{3.17}$$

Constraint (3.18) ensures that the time spent for a single successful transmission will be smaller than the allowable delay from node i to relay node κ. We assume that queuing delay is ignored, the delay over link (i,k) is the time spent for a single successful, which is the number of transmission before getting the first success $\left[\frac{1}{q_i(1-q_i)^{n-1}q_\kappa P_{i\kappa}(r_i,m)} - 1\right]$ times the timeout interval τ_i for each failure transmission plus one slot time for the success transmission. It is expressed as

$$\tau_i \left[\frac{1}{q_i(1-q_i)^{n-1} q_\kappa P_{i\kappa}(r_i, m)} - 1 \right] + 1 \leq \overline{T_{i\kappa}} \qquad \forall i \in S, m \in M \tag{3.18}$$

Constraint (3.19) and (3.20) ensures that the value of q_i, q_κ and $P_{i\kappa}(r_i, m)$ fall within a small number ϵ and 1. Since q_i, q_κ and $P_{ij}(r_i, m)$ denotes the probability that node i is active in a time slot, the probability that node j is active in a time slot and the probability of node i to transmit packet to node j without error with transmission range radius of r_i, respectively, it is required that q_i, q_κ and $P_{i\kappa}(r_i, m)$ fall in between a small number ϵ and 1. The constraints are expressed as :

$$\epsilon \leq q_i, q_\kappa \leq 1 \qquad \forall i \in S \tag{3.19}$$

$$\epsilon \leq P_{i\kappa}(r_i, m) \leq 1 \qquad \forall i \in S, m \in M \tag{3.20}$$

Constraint (3.21) and (3.22) are expressed as

$$r_i \in R_i \qquad \forall i \in S \tag{3.21}$$

doi:10.6342/NTU202210101

$$r_\kappa \in R_\kappa \tag{3.22}$$

As for the definition of when the sensor network will paralyzed or is considered unfunctional, we assume that when a single node is out of batteries, the wireless sensor network will be considered as not working. From the energy consumption of each node and the probability of being active, the time that a node can function can be calculated. We assume that there is a goal of system life time that needs to acheive. In order to achieve this goal, a conditional limit can be listed by the initial power of each sensor node to ensure that every node can achieve the goal of the system life time. We define formula (3.23) and (3.24)below to make sure the life time of the nodes will exceed the expected life time of the sensor network.

$$\left[C_i^a(r_i, m)q_i + C_i^b(r_i, m)(1 - q_i) + C_i^\tau(r_i, m)q_i\right] t \leq P_i \qquad \forall i \in S \tag{3.23}$$

$$\left[C_\kappa^a(r_\kappa, m)q_\kappa + C_k^b(r_\kappa, m)(1 - q_\kappa)\right] t \leq P_\kappa \tag{3.24}$$

3.4 Model 3 : Network Tree Structure Relationship

The given parameters as well as their descriptions are shown in Table 3.5:

doi:10.6342/NTU202210101

Table 3.5: Given Parameters

Notation	Description
N	The index set of all possible numbers of subtrees, which is $\{1, 2, 3, \ldots, \overline{n}\}$
V	The index set of all possible numbers of sensor nodes that a single subtree contains, which is $\{1, 2, 3, \ldots, \overline{v}\}$
θ_{ij}	The number $j \in V$ sensor node from subtree $i \in N$, responsible for sensing and gathering data from different area
κ_i	The relay node from subtree $i \in N$, responsible for aggregating data sent from the lower level layer sensor nodes
ξ	The sink node, responsible for aggregating data sent from the relay nodes
$\overline{T_{\theta_i,\xi}}$	The allowable end to end delay from sensor node to sink node for subtree $i \in N$
d_{θ_i,κ_i}	The distance between sensor nodes and relay node from subtree $i \in N$
$R_{\theta i}$	Set of possible range for sensor nodes in subtree $i \in N$
R_κ	Set of possible range for relay node in subtree $i \in N$
R_ξ	Set of possible range for sink node
τ_{θ_i}	Timeout interval for sensor nodes from subtree $i \in N$ (a given $\sharp$ of time slots)
τ_{κ_i}	Timeout interval for relay nodes from subtree $i \in N$ (a given $\sharp$ of time slots)
t_{θ_i,κ_i}	The transmission time from sensor nodes to relay node , e.g. 1 slot time
t	The least expected time slots for the network to work
P_i	The initial power storage for sensor node $i \in S$
P_{κ_i}	The initial power storage for relay node from subtree $i \in N$
P_ξ	The initial power storage for sink node ξ

The decision variables and their descriptions are are shown in Table 3.6:

doi:10.6342/NTU202210101

Also the functions $C^a_{\theta_i}(r_{\theta i}, m), C^b_{\theta_i}(r_{\theta i}, m), C^a_{\kappa_i}(r_{\kappa_i}, m), C^b_{\kappa_i}(r_{\kappa_i}, m), C^a_{\xi}(r_{\xi}, m), C^b_{\xi}(r_{\xi}, m)$, $C^{\tau}_{\theta_i}(r_{\theta i}, m)$ and $C^{\tau}_{\kappa_i}(r_{\kappa_i}, m)$ are given, the parameters of the function are to be determined, so the value of the function itself is a decision variable.

The next step we consider the whole network tree structure with sensor nodes denoted as θ, relay nodes denoted as κ and sink node denoted as ξ. This Network Tree Structure Relationship (Model 3)is an extension of Model 2 introduced in the previous chapter. The tree structure is shown as Figure 3.3.

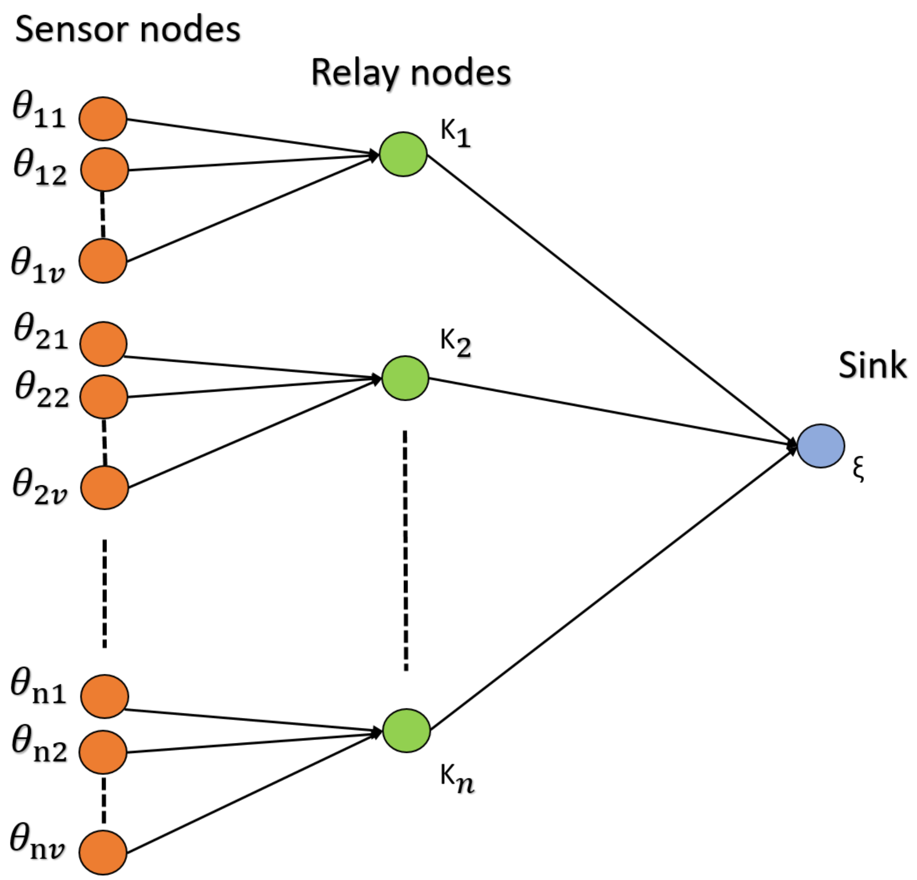

Figure 3.3: Network Tree Structure Relationship

We assume that the network tree structure is a data-centric network, and we take all subtrees into consideration. Each subtree contains of v amount of sensor nodes and a relay

doi:10.6342/NTU202210101

Table 3.6: Decision Variables

Notation	Description
$r_{\theta i}$	Transmission range of sensor nodes in subtree $i \in N$,$r_{\theta i} \in R_{\theta i}$
r_{κ_i}	Transmission range of relay node in subtree $i \in N$,$r_{\kappa_i} \in r_{\kappa_i}$
r_{ξ}	Transmission range of sink node
M	The index set of all possible packet size , which is $\{1, 2, 3, \ldots, \overline{m}\}$
q_{θ_i}	The probability that sensor nodes in subtree $i \in N$ is active in a time slot
$q_{\kappa R_i}$	The probability that relay node in subtree $i \in N$ is active to recieve data in a time slot
$q_{\kappa S_i}$	The probability that relay node in subtree $i \in N$ is active to send data a time slot
q_{ξ}	The probability that sink node is active in a time slot
$P_{\theta_i,\kappa_i}(r_{\theta i}, m)$	The probability of sensor nodes to transmit packet with $m \in M$ size to relay node in subtree $i \in N$ when no error occurs with transmission range radius of $r_{\theta i} \in R_{\theta i}$
$P_{\kappa_i,\xi}(r_{\kappa_i}, m)$	The probability of the relay node in subtree $i \in N$ to transmit packet with $m \in M$ size to sink node when no error occurs with transmission range radius of $r_{\kappa_i} \in r_{\kappa_i}$
$C^a_{\theta_i}(r_{\theta i}, m)$	The average power consumption rate when sensor nodes in subtree $i \in N$ is active with transmission range $r_{\theta i} \in R_{\theta i}$ in 1 time slot(influenced by m)
$C^b_{\theta_i}(r_{\theta i}, m)$	The average power consumption rate when sensor nodes in subtree $i \in N$ is inactive with transmission range $r_{\theta i} \in R_{\theta i}$ in 1 time slot(influenced by m)
$C^a_{\kappa_i}(r_{\kappa_i}, m)$	The average power consumption rate when the relay node in subtree $i \in N$ is active with transmission range $r_{\kappa_i} \in r_{\kappa_i}$ in 1 time slot(influenced by m)
$C^b_{\kappa_i}(r_{\kappa_i}, m)$	The average power consumption rate when the relay node in subtree $i \in N$ is inactive with transmission range $r_{\kappa_i} \in r_{\kappa_i}$ in 1 time slot(influenced by m)
$C^a_{\xi}(r_{\xi}, m)$	The average power consumption rate when the sink node is active with transmission range r_{ξ} in 1 time slot(influenced by m)
$C^b_{\xi}(r_{\xi}, m)$	The average power consumption rate when the sink is inactive with transmission range r_{ξ} in 1 time slot(influenced by m)
$C^{\tau}_{\theta_i}(r_{\theta i}, m)$	The average power consumption rate for sensor nodes in subtree $i \in N$ to transmit a $m \in M$ size packet(in 1 time slot) with transmission range $r_{\theta i} \in R_{\theta i}$
$C^{\tau}_{\kappa_i}(r_{\kappa_i}, m)$	The average power consumption rate for the relay node in subtree $i \in N$ to transmit a $m \in M$ size packet(in 1 time slot) with transmission range $r_{\kappa_i} \in r_{\kappa_i}$

doi:10.6342/NTU202210101

node. There are n subtrees and a sink node that comprises into a whole tree structure. A single subtree is shown as Figure 3.4. We assume that based on fairness, sensor nodes in a same subtree has the same probability to be active shown as below.

$$q_{\theta_{11}} = q_{\theta_{12}} = ... = q_{\theta_{1v}} = q_{\theta_1} \tag{3.25}$$

Due to different monitor areas of different sensor nodes, they can have different importance and priorities. The goal is to aggregate data from all sensors into the sink in minimum energy consumption, while satisfied to the average possible delay constraints in all possible origin-destination pairs. This delay should be the maximum link delay of all possible links.

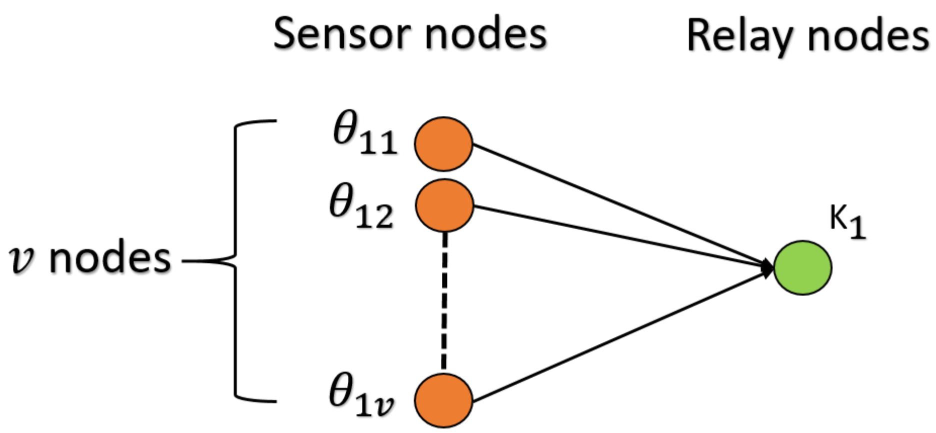

Figure 3.4: Subtree 1

Sensor nodes are responsible for sensing purpose, therefore their job is to send data to relay nodes, and we assume that they will send data as soon as they are active. Relay nodes are responsible for both sending and receiving data. They will aggregate data from sensor nodes and then transmit them to the sink. It is also assumed that relay nodes will send data as soon as they are active and also receive data when they are active. Sink node

doi:10.6342/NTU202210101

is for aggregating data from relay nodes, so in this model its only purpose is to receive data. In the scenario we consider, we assume that each point is parallel and competing for an interface in a slotted time. Every sensor has information to send and will send when there is a chance.

doi:10.6342/NTU202210101

The objective function is expressed as formula (3.26):

$$\min \Bigg[\Big(C^a_{\theta_1}(r_{\theta_1},m)q_{\theta_1}v + C^a_{\theta_2}(r_{\theta_2},m)q_{\theta_2}v + ... + C^a_{\theta_n}(r_{\theta_n},m)q_{\theta_n}v$$

$$+ C^b_{\theta_1}(r_{\theta_1},m)(1-q_{\theta_1})v + C^b_{\theta_2}(r_{\theta_2},m)(1-q_{\theta_2})v + ... + C^b_{\theta_n}(r_{\theta_n},m)(1-q_{\theta_n})v$$

$$+ C^\tau_{\theta_1}(r_{\theta_1},m)q_{\theta_1} \times v + C^\tau_{\theta_2}(r_{\theta_2},m)q_{\theta_2} \times v + ... + C^\tau_{\theta_n}(r_{\theta_n},m)q_{\theta_n} \times v\Big)$$

$$+ \Big(C^a_{\kappa_1}(r_{\kappa_1},m)q_{\kappa R_1} + C^a_{\kappa_2}(r_{\kappa_2},m)q_{\kappa R_2} + ... + C^a_{\kappa_n}(r_{\kappa_n},m)q_{\kappa R_n}$$

$$+ C^b_{\kappa_1}(r_{\kappa_1},m)(1-q_{\kappa R_1}) + C^b_{\kappa_2}(r_{\kappa_2},m)(1-q_{\kappa R_2}) + ... + C^b_{\kappa_n}(r_{\kappa_n},m)(1-q_{\kappa R_n})$$

$$+ C^\tau_{\kappa_1}(r_{\kappa_1},m)q_{\kappa S_1} + C^\tau_{\kappa_2}(r_{\kappa_2},m)q_{\kappa S_2} + ... + C^\tau_{\kappa_n}(r_{\kappa_n},m)q_{\kappa S_n}\Big)$$

$$+ C^a_\xi(r_\xi,m)q_\xi + C^b_\xi(r_\xi,m)(1-q_\xi)\Bigg] \tag{3.26}$$

Based on fairness, let each subtree contain the same probability of activeness. Therefore we can also consider formula (3.27) ,formula (3.28) and formula (3.29) and we then consider sensor nodes in the same subtree i as θ_i:

$$q_{\theta_1} = q_{\theta_2} = q_{\theta_3} = ... = q_{\theta_n} \tag{3.27}$$

$$q_{\kappa R_1} = q_{\kappa R_2} = q_{\kappa R_3} = ... = q_{\kappa R_n} \tag{3.28}$$

$$q_{\kappa S_1} = q_{\kappa S_2} = q_{\kappa S_3} = ... = q_{\kappa S_n} \tag{3.29}$$

The new objective function based on formula (3.27) and formula (3.28) is shown as formula (3.30) and we also consider the packet size level that node i is transmitting because it would influence the size of the time slot. Therefore, the bigger the packet size, the longer the time slot will be needed to transmit a packet. The probability of transmitting a packet without error will also decrease when m increases, and we assume each packet

doi:10.6342/NTU202210101

has a fixed size header, therefore the larger the packet size is, the larger the throughput is. We take all the above mentioned factors into consideration to find the trade off of the packet size and the power consumption in a single byte. So the objective function will be divided by m to normalize by the length of time slot. :

$$\min \quad \sum_{i \in N} \Bigg[\frac{C^a_{\theta_i}(r_{\theta_i}, m) q_{\theta_i} v + C^b_{\theta_i}(r_{\theta_i}, m)(1 - q_{\theta_i}) v + C^{\tau}_{\theta_i}(r_{\theta_i}, m) q_{\theta_i} v}{m}$$
$$+ \frac{C^a_{\kappa_i}(r_{\kappa_i}, m) q_{\kappa R_i} + C^b_{\kappa_i}(r_{\kappa_i}, m)(1 - q_{\kappa R_i}) + C^{\tau}_{\kappa_i}(r_{\kappa_i}, m) q_{\kappa S_i}}{m}$$
$$+ \frac{C^a_{\xi}(r_{\xi}, m) q_{\xi} + C^b_{\xi}(r_{\xi}, m)(1 - q_{\xi})}{m} \Bigg] \tag{3.30}$$

Constraint (3.31) ensures that the time spent for a single successful transmission from sensor nodes to sink node ξ will be smaller than the allowable delay. We assume that queuing delay is ignored, the time spent from sink node to relay node is the time spent for a single successful, which is the number of transmission before getting the first success $\left[\frac{1}{q_{\theta_i}(1-q_{\theta_i})^{(v-1)} q_{\kappa R_i} P_{\theta_i,\kappa_i}(r_{\theta_i}, m)} - 1\right]$ times the timeout interval for each transmission τ_{θ_i} and plus one slot time for the success transmission. And we also have to add the time spent from relay node to sink node, which is $\left[\frac{1}{q_{\kappa S_i}(1-q_{\kappa S_i})^{(n-1)} q_{\xi} P_{\kappa_i,\xi}(r_{\kappa_i}, m)} - 1\right]$ times the timeout interval for each transmission τ_{κ_i} and plus one slot time for the success transmission. It is expressed as

$$\tau_{\theta_i}\left[\frac{1}{q_{\theta_i}(1-q_{\theta_i})^{(v-1)} q_{\kappa R_i} P_{\theta_i,\kappa_i}(r_{\theta_i}, m)} - 1\right] + \tau_{\kappa_i}\left[\frac{1}{q_{\kappa S_i}(1-q_{\kappa S_i})^{(n-1)} q_{\xi} P_{\kappa_i,\xi}(r_{\kappa_i}, m)} - 1\right] + 2 \le \overline{T_{\theta_i,\xi}}$$
$$\forall i \in N, m \in M \tag{3.31}$$

Constraint (3.34) ensures that the output throughput of the relay node is greater than the

doi:10.6342/NTU202210101

input throughput so that overflowing won't occur. This constraint is a distinguishing feature compared to Model 2. In Model 2, we assume that sensor nodes always have data to send and is constantly collecting data. Therefore we assume that there is an internal flow control mechanism for avoiding overflow in sensor nodes. That is to mark those data that stays for too long and untransmitted as obsolete, so as to avoid overflowing in sensor nodes. In model 3, there exist relay nodes and were responsible for transferring data collected from the sensor nodes. Hence, relay nodes don't have the ability to discard packets, and that's what make the throughput constraint important. We assume that relay nodes have a certain amount of capacity to temporarily store those data that were unable to be send and they will always send when there is a chance. Thus, the average output throughput should be greater or equal to the average output throughput in order to avoid buffer overflow.

Constraint (3.34) consists of two parts, the input throughput and the output throughput. The input throughput of the relay node is composed of the summation output throughput of all sensor nodes, which is the size of the packet sent by sensor nodes divided by the amount of time slots needed for a single successful transmit to pass the data to relay node times the amount of sensor nodes. It is shown as formula (3.32)

$$m \div \frac{1}{q_{\theta_i}(1-q_{\theta_i})^{(v-1)} q_{\kappa R_i} P_{\theta_i,\kappa_i}(r_{\theta_i}, m)} \times v \tag{3.32}$$

The output throughput of the relay node is the size of the packet sent by relay node

doi:10.6342/NTU202210101

divided by the amount of time slots needed for a single successful transmit to pass the data to sink node.It is shown as formula (3.33)

$$m \div \frac{1}{q_{\kappa S_i}(1-q_{\kappa S_i})^{(n-1)} q_\xi P_{\kappa_i,\xi}(r_{\kappa_i}, m)} \tag{3.33}$$

The complete throughput constraint is expressed as

$$m q_{\kappa S_i}(1-q_{\kappa S_i})^{(n-1)} q_\xi P_{\kappa_i,\xi}(r_{\kappa_i}, m) \geq m v q_{\theta_i}(1-q_{\theta_i})^{(v-1)} q_{\kappa R_i} P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$$

$$\forall i \in N, m \in M \tag{3.34}$$

Constraint (3.35) ,(3.36),(3.37),(3.38) and (3.39) ensures that the value of q_{θ_i} ,$q_{\kappa R_i}$, $q_{\kappa S_i}$, q_ξ , $P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ and $P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ fall within a small number ϵ and 1. Since q_{θ_i} ,$q_{\kappa R_i}$, $q_{\kappa S_i}$, q_ξ , $P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ and $P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ denotes the probability that node θ_i is active in a time slot, the probability that node κ_i is active for receiving in a time slot, the probability that node κ_i is active for sending in a time slot, the probability that node ξ is active in a time slot, the probability of node θ_i to transmit packet to node κ_i without error with transmission range radius of r_{θ_i} and the probability of node κ_i to transmit packet to node ξ without error with transmission range radius of r_{κ_i}, respectively, it is required that q_{θ_i} ,$q_{\kappa R_i}$, $q_{\kappa S_i}$, q_ξ , $P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ and $P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ fall in between a small number ϵ and 1. The constraints are expressed as :

$$\epsilon \leq q_{\theta_i} \leq 1 \qquad \forall i \in N \tag{3.35}$$

$$\epsilon \leq q_{\kappa R_i}, q_{\kappa S_i} \leq 1 \qquad \forall i \in N \tag{3.36}$$

$$\epsilon \leq q_\xi \leq 1 \tag{3.37}$$

doi:10.6342/NTU202210101

$$\epsilon \leq P_{\theta_i,\kappa_i}(r_{\theta_i}, m) \leq 1 \qquad \forall i \in N, m \in M \tag{3.38}$$

$$\epsilon \leq P_{\kappa_i,\xi}(r_{\kappa_i}, m) \leq 1 \qquad \forall i \in N, m \in M \tag{3.39}$$

Constraint (3.40) ,(3.41)and (3.42) are expressed as

$$r_{\theta_i} \in R_{\theta_i} \qquad \forall i \in N \tag{3.40}$$

$$r_{\kappa_i} \in R_{\kappa_i} \qquad \forall i \in N \tag{3.41}$$

$$r_{\xi} \in R_{\xi} \tag{3.42}$$

As for the definition of when the sensor network will paralyzed or is considered unfunctional, we assume that when a single node is out of batteries, the wireless sensor network will be considered as not working. From the energy consumption of each node and the probability of being active, the time that a node can function can be calculated. We assume that there is a goal of system life time that needs to acheive. In order to achieve this goal, a conditional limit can be listed by the initial power of each sensor node to ensure that every node can achieve the goal of the system life time. We define formula (3.43), (3.44) and (3.45)below to make sure the life time of the nodes will exceed the expected life time of the sensor network.

$$\left[C^a_{\theta_i}(r_{\theta_i}, m)q_{\theta_i} + C^b_{\theta_i}(r_{\theta_i}, m)(1 - q_{\theta_i}) + C^\tau_{\theta_i}(r_{\theta_i}, m)q_{\theta_i}\right] t \leq P_{\theta_i} \qquad \forall i \in N \tag{3.43}$$

$$\left[C^a_{\kappa_i}(r_{\kappa_i}, m)q_{\kappa R_i} + C^b_{\kappa_i}(r_{\kappa_i}, m)(1 - q_{\kappa R_i}) + C^\tau_{\kappa_i}(r_{\kappa_i}, m)q_{\kappa S_i}\right] t \leq P_{\kappa_i} \quad \forall i \in N \tag{3.44}$$

$$\left[C^a_{\xi}(r_{\xi}, m)q_{\xi} + C^b_{\xi}(r_{\xi}, m)(1 - q_{\xi})\right] t \leq P_{\xi} \tag{3.45}$$

doi:10.6342/NTU202210101

Chapter 4 Solution Approach

In this chapter, the solution approach for the optimization-based power controlling problem described in the previous chapter will be thoroughly explained. As described in Chapter 3, the mathematical model of this problem is very intricate. Lagrangian Relaxation is a mature and widely used approach in this kind of complex constrained optimization problem. We will explain its solution procedure thoroughly in the next section and illustrate how the problem is solved using Lagrangian Relaxation.

4.1 Lagrangian Relaxation Method

Lagrangian Relaxation(LR) is a tool that is widely used in mathematical programming applications. It was first proposed in the 1970s to solve general mixed integer programs with ”complicated” constraints [42]. There are constraints that can be simply solved and constraints that are difficult to solve or require exponential time to solve it. The concept of Lagrangian Relaxation is to relax those constraints that are difficult to solve [43].

doi:10.6342/NTU202210101

By relaxing the constraints that are difficult to solve, the primal problem will then become a Lagrangian Relaxation problem which is relatively easy to solve.

We create this Lagrangian Relaxation problem by moving the difficult constraints to the primal objective function with respective coefficients, namely Lagrangian multipliers. Lagrangian multipliers are considered penalties when constraints are violated [44]. After relaxing all constraints that are considered difficult to solve, the LR problem can be decomposed into several subproblems. By dividing into subproblems, it will be much easier to solve. Subproblems are decomposed from the LR problem by separating the constraints and the part of LR problems that contains the same decision variables. Each subproblem will then be solved optimally by using the concept of divide and conquer. Also the subproblems will be solved according to their characteristics. We can then obtain the solution to the LR problem with the regarding decision variable.

With the solution we obtained from the LR problem, it will form a lower bound(LB) in a minimization problem. And if the solution is feasible to the primal objective function, which means that it does not violate any constraints, an Upper bound(UB) will appear. The primal optimal solution will be bounded by the lower bound and the upper bound. If the solution to the LR problem is not feasible, we should tune the solution by heuristic methods in order to convert the solution to a feasible solution.

The Lagrangian multipliers must be adjusted in order to find the tightest lower bound.

doi:10.6342/NTU202210101

The Subgradient method is a commonly used method when solving the dual problem. After updating the Lagrangian multipliers, we will then solve the subproblems again to gradually decrease the gap between the lower bound(LB) and upper bound(UB). This process will continue to proceed until it meets the conditions such as iteration limit or when the gap between the lower bound and upper bound is less than a specific threshold. The gap between the lower bound and upper bound can also help us measure the quality of the solution. Smaller gaps indicates better solutions, and when the lower bound overlaps with the upper bound, the best solution appears. Figure 4.1 shows the procedure of the Lagrangian Relaxation method mentioned in this section.

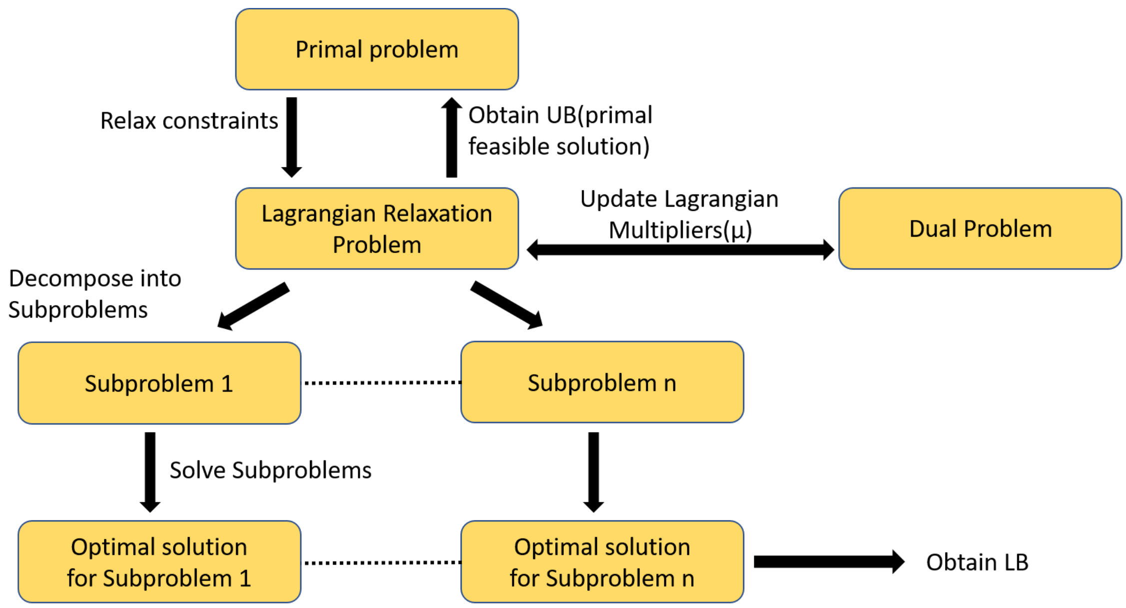

Figure 4.1: The procedure of Lagrangian Relaxation

doi:10.6342/NTU202210101

4.2 Model 1 : One-to-One Relationship

4.2.1 Deal with Decision Variables

Constraint (3.5) contains of the product of decision variables $(q_i \times q_j \times P_{ij}(r_i, m))$. With these three decision variables multiplying, two issues will occur. First, the convexity of the function is being destroyed . Second, decision variables will not be able to ” separate” or ”decomposed” when solving it. In order to deal with it,we let

$$q_i \times q_j \times P_{ij}(r_i, m) = Z_{ij} \qquad \forall i \in S, j \in S, m \in M \tag{4.1}$$

Apply logarithmic operation on both sides,

$$\Rightarrow \log q_i + \log q_j + \log P_{ij}(r_i, m) = \log Z_{ij} \qquad \forall i \in S, j \in S, m \in M \tag{4.2}$$

so that we can avoid decision variables from multiplying.

We derived

$$\Rightarrow \epsilon^3 \le Z_{ij} \le 1 \qquad \forall i \in S, j \in S \tag{4.3}$$

from formulation (4.2).

After the transposition in constraint (3.5) with the replacement of $q_i \times q_j \times P_{ij}(r_i, m)$ to Z_{ij}, we derive the formulation

$$\frac{1}{Z_{ij}} \le 1 + \frac{1}{\tau_i}(\overline{T_{ij}} - 1) \qquad \forall i \in S, j \in S \tag{4.4}$$

Let

doi:10.6342/NTU202210101

$$\frac{1}{Z_{ij}} \leq 1 + \frac{1}{\tau_i}(\overline{T_{ij}} - 1) = \alpha_{ij} \qquad \forall i \in S, j \in S \tag{4.5}$$

and derive

$$\Rightarrow \frac{1}{\alpha_{ij}} \leq Z_{ij} \qquad \forall i \in S, j \in S \tag{4.6}$$

which max { ϵ^3, $\frac{1}{\alpha_{ij}}$ } is the underbound of Z_{ij}.

A LR problem is derived as formula (4.7) and we can later apply to Lagrangian Relaxation:

$$\min \quad \sum_{i \in S} \sum_{j \in S} \frac{\left[C_i^a(r_i, m)q_i + C_j^a(r_j, m)q_j + C_i^b(r_i, m)(1 - q_i) + C_j^b(r_j, m)(1 - q_j) + C_i^\tau(r_i, m)q_i\right]}{m}$$
$$+ \sum_{i \in S} \sum_{j \in S} \mu_{ij}^1(\log q_i + \log q_j + \log P_{ij}(r_i, m) - \log Z_{ij}) \tag{4.7}$$

where μ_{ij}^1 can be negative or positive.

The objective function is subject to :

$$\epsilon \leq q_i \leq 1 \qquad \forall i \in S \tag{4.8}$$

$$\epsilon \leq q_j \leq 1 \qquad \forall j \in S \tag{4.9}$$

$$\epsilon \leq P_{ij}(r_i, m) \leq 1 \qquad \forall i \in S, j \in S, m \in M \tag{4.10}$$

$$\max\{\epsilon^3, \frac{1}{\alpha_{ij}}\} \leq Z_{ij} \leq 1 \qquad \forall i \in S, j \in S \tag{4.11}$$

$$r_i \in R_i \qquad \forall i \in S \tag{4.12}$$

If $\lambda_{ij} << 1$, which makes $\frac{1}{\lambda_{ij}}$ a great number, we can see this as a M/G/1 system where transmission time is a random variable governed by a geometric distribution.

doi:10.6342/NTU202210101

$$\Rightarrow Prof:\{k \quad retransmission\} = (1-Z_{ij})^k Z_{ij} \quad \forall i \in S, j \in S, m \in M \quad (4.13)$$

Each retransmission will cost us a timeout interval of τ_i, and the last transmission that succeed will cost a time slot. Based on these assumption we can consider queuing delay in variable αij.

If r_i is a continuous variable then $C_i^a(r_i, m)$ will become a decision variable and we might also need to deal with $C_i^a(r_i, m) \times q_i$.

In model one (One-to-One Relationship), we can consider r_i as a fixed number with limited amount of values in different situations. But it won't be a good solution when model one is extended to model two (Many-to-One Relationship) and model three (Network Tree Structure Relationship), so we will try to solve this problem in this section.

In model two (Many-to-One Relationship) and model three (Network Tree Structure Relationship), there are many-to-one relationships with hub and aggregation of data flow. If we consider r_i as a fixed number with limited amount of values as we did in model one (One-to-One Relationship), there occurs a problem called combinatorial explosion. A combinatorial explosion happens when the combinatorics of the problem is affected by the input, constraints, and bounds of the problem which leads to the rapid growth of the complexity of a problem.

Because of the combinatorial explosion, we can no longer see r_i as a fixed number with limited amount of values, we have to reformulate the objective function to keep de-

doi:10.6342/NTU202210101

cision variables from multiplying. We expand formula (4.7) and get formula (4.14) as follows.

$$\min \quad \sum_{i\in S}\sum_{j\in S}\Bigg(\frac{C_i^a(r_i,m)q_i + C_j^a(r_j,m)q_j + C_i^b(r_i,m) - C_i^b(r_i,m)q_i + C_j^b(r_i,m) - C_j^b(r_j,m)q_j}{m}$$

$$+ \frac{C_i^\tau(r_i,m)q_i}{m}\Bigg)$$

$$+ \sum_{i\in S}\sum_{j\in S}\mu_{ij}^1(\log q_i + \log q_j + \log P_{ij}(r_i,m) - \log Z_{ij}) \tag{4.14}$$

In order to deal with $C_i^a(r_i,m)\times q_i, C_j^a(r_j,m)\times q_j, C_i^b(r_i,m)\times q_i$ and $C_j^b(r_j,m)\times q_j$, we let

$$C_i^a(r_i,m)\times q_i = x_i \qquad \forall i\in S \tag{4.15}$$

$$C_j^a(r_j,m)\times q_j = x_j \qquad \forall j\in S \tag{4.16}$$

$$C_i^b(r_i,m)\times q_i = y_i \qquad \forall i\in S \tag{4.17}$$

$$C_j^b(r_j,m)\times q_j = y_j \qquad \forall j\in S \tag{4.18}$$

Apply logarithmic operation on both sides,

$$\Rightarrow \log C_i^a(r_i,m) + \log q_i = \log x_i \qquad \forall i\in S \tag{4.19}$$

$$\Rightarrow \log C_j^a(r_j,m) + \log q_j = \log x_j \qquad \forall j\in S \tag{4.20}$$

$$\Rightarrow \log C_i^b(r_i,m) + \log q_i = \log y_i \qquad \forall i\in S \tag{4.21}$$

$$\Rightarrow \log C_j^b(r_j,m) + \log q_j = \log y_j \qquad \forall j\in S \tag{4.22}$$

A new LR function is derived as the formula below:

doi:10.6342/NTU202210101

$$
\begin{aligned}
\min \quad & \sum_{i \in S} \sum_{j \in S} \frac{\left(x_i + x_j + C_i^b(r_i, m) - y_i + C_j^b(r_j, m) - y_j + C_i^\tau(r_i, m) q_i\right)}{m} \\
& + \sum_{i \in S} \sum_{j \in S} \mu_{ij}^1 (\log q_i + \log q_j + \log P_{ij}(r_i, m) - \log Z_{ij}) \\
& + \sum_{i \in S} \mu_i^2 (\log C_i^a(r_i, m) + \log q_i - \log x_i) \\
& + \sum_{j \in S} \mu_j^3 (\log C_j^a(r_j, m) + \log q_j - \log x_j) \\
& + \sum_{i \in S} \mu_i^4 (\log C_i^b(r_i, m) + \log q_i - \log y_i) \\
& + \sum_{j \in S} \mu_j^5 (\log C_j^b(r_j, m) + \log q_j - \log y_j)
\end{aligned}
\tag{4.23}
$$

There still exists decision variables $C_i^\tau(r_i, m)$ and q_i multiplying, and we also have to deal with it.

Let

$$
C_i^\tau(r_i, m) q_i = \beta_{ij} \qquad \forall i \in S, m \in M \tag{4.24}
$$

Apply logarithmic operation on both sides,

$$
\Rightarrow \log C_i^\tau(r_i, m) + \log q_i = \log \beta_{ij} \qquad \forall i \in S, m \in M \tag{4.25}
$$

Multiply both side with $C_i^\tau(r_i, m)$ and we obtain constraint(4.26)

$$
\epsilon \times C_i^\tau(\min R_i, m) \leq \beta_{ij} \leq C_i^\tau(\max R_i, m) \qquad \forall i \in S, m \in M \tag{4.26}
$$

We relax formula (4.25) into the LR problem formula (4.23).

doi:10.6342/NTU202210101

The final form of LR problem in model 1(One-to-One Relationship) is shown as formula (4.27):

$$
\begin{aligned}
\min \quad & \sum_{i \in S} \sum_{j \in S} \frac{\left(x_i + x_j + C_i^b(r_i, m) - y_i + C_j^b(r_j, m) - y_j + \beta_{ij}\right)}{m} \\
& + \sum_{i \in S} \sum_{j \in S} \mu_{ij}^1 (\log q_i + \log q_j + \log P_{ij}(r_i, m) - \log Z_{ij}) \\
& + \sum_{i \in S} \mu_i^2 (\log C_i^a(r_i, m) + \log q_i - \log x_i) \\
& + \sum_{j \in S} \mu_j^3 (\log C_j^a(r_j, m) + \log q_j - \log x_j) \\
& + \sum_{i \in S} \mu_i^4 (\log C_i^b(r_i, m) + \log q_i - \log y_i) \\
& + \sum_{j \in S} \mu_j^5 (\log C_j^b(r_j, m) + \log q_j - \log y_j) \\
& + \sum_{i \in S} \sum_{j \in S} \mu_{ij}^6 (\log C_i^\tau(r_i, m) + \log q_i - \log \beta_{ij})
\end{aligned}
\tag{4.27}
$$

The objective function is subject to :

$$\epsilon \leq q_i \leq 1 \qquad \forall i \in S \tag{4.28}$$

$$\epsilon \leq q_j \leq 1 \qquad \forall j \in S \tag{4.29}$$

$$\epsilon \leq P_{ij}(r_i, m) \leq 1 \qquad \forall i \in S, j \in S, m \in M \tag{4.30}$$

$$\max\{\epsilon^3, \frac{1}{\alpha_{ij}}\} \leq Z_{ij} \leq 1 \qquad \forall i \in S, j \in S \tag{4.31}$$

$$\epsilon^2 \leq x_i \leq C_i^a(r_i, m) \qquad \forall i \in S \tag{4.32}$$

$$\epsilon^2 \leq x_j \leq C_j^a(r_j, m) \qquad \forall j \in S \tag{4.33}$$

$$\epsilon^2 \leq y_i \leq C_i^b(r_i, m) \qquad \forall i \in S \tag{4.34}$$

doi:10.6342/NTU202210101

$$\epsilon^2 \leq y_j \leq C_j^b(r_j, m) \qquad \forall j \in S \qquad (4.35)$$

$$r_i \in R_i \qquad \forall i \in S \qquad (4.36)$$

$$\epsilon \times C_i^\tau(\min R_i, m) \leq \beta_{ij} \leq C_i^\tau(\max R_i, m) \qquad \forall i \in S, j \in S, m \in M \qquad (4.37)$$

Multipliers $\mu^1, \mu^2, \mu^3, \mu^4, \mu^5and\mu^6$respectively represents the vectors of $\{\mu_{ij}^1\}$, $\{\mu_i^2\}$, $\{\mu_j^3\}$, $\{\mu_i^4\}$, $\{\mu_j^5\}$ and $\{\mu_{ij}^6\}$. These multipliers are either positive or negative due to relaxing equality constraints.

With the objective function and the constraints above, we can proceed to cut q_i , q_j and $P_{ij}(r_i, m)$ into 1000 pieces of equal segments. By using exhaust search we can eventually exploit all possible solutions and try to get the primal solution.

4.2.2 The LR Subproblems

To solve this Lagrangian Relaxation problem easily and effectively, we can divide the problem into subproblems and solve them respectively. The way of decomposing this Lagrangian Relaxation problem is by separating the decision variables. Each decision variable will form a subproblem and because the original problem is a minimization problem, the subproblems will also be a minimization problem. Each subproblems will then be dealt with different solution approaches base on their characteristics. This divide and conquer technique will be introduce and implement in this section.

doi:10.6342/NTU202210101

4.2.2.1 Subproblem 1(related to decision variable x_i)

By extracting all decision variables x_i in the LR problem, we will then obtain subproblem 1. In subproblem 1, μ_i^2 can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 1, we have to take the sign of μ_i^2 into consideration. Linear term will not influence the concavity or convexity of the formula, so μ_i^2 will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ_i^2 is positive with a negative sign in front of it, the formula will become a convex function. And when μ_i^2 is negative, the formula will become a concave function.

The concavity or convexity of subproblem 1 will lead to different solution approaches. When μ_i^2 is positive and therefore changing the coefficient of log term into negative, find the point of x_i where the slope is 0, which will be the minimum point of the convex function. If the point of x_i where the slope is 0 falls in the legal range, which is the upper and lower bound of x_i, then it is the solution to x_i. However, if this point does not fall in the legal range of x_i, the upper and lower bound of x_i will be substitute into the objective function to see whichever is smaller, and it will be the solution to x_i.

On the contrary, when μ_i^2 is negative and therefore changing the coefficient of log

doi:10.6342/NTU202210101

term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to x_i.

The minimum of x_i occurs when both $C_i^a(r_i, m)$ and q_i are at its minimum, which is ϵ. The maximum on the other hand occurs when q_i is 1. We separate x_i from the objective function and derive a subproblem shown as formula (4.38) and constraint shown as formula (4.39).

Table 4.1: Subproblem 1(related to decision variable x_i)

Objective function :

$$Z_{sub1} = \min \sum_{i \in S} (\frac{x_i}{m} - \mu_i^2 \log x_i) \tag{4.38}$$

Subject to :

$$\epsilon^2 \leq x_i \leq C_i^a(r_i, m) \qquad \forall i \in S \tag{4.39}$$

doi:10.6342/NTU202210101

Algorithm 1: Algorithm for Subproblem 1

for *each node i* **do**
  Calculate the objective value at $x_i = \epsilon^2$ denoted as V_1
  Calculate the objective value at $x_i = C_i^a(r_i, m)$ denoted as V_2
  if $\mu_i^2 \geq 0$ **then**
    partial differential to x_i
    $\frac{1}{m} - \mu_i^2 \frac{1}{x_i \ln e} = 0$
    $x_i = \mu_i^2 m$
    if x_i *is not feasible* **then**
      set x[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set x[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.2.2.2 Subproblem 2(related to decision variable x_j)

The procedure of dividing x_j from the LR problem is similar with x_i. By extracting all decision variables x_j in the LR problem, we will then obtain subproblem 2. In subproblem 2, μ_j^3 can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 2, we have to take the sign of μ_j^3 into consideration. Linear term will not influence the concavity or convexity of the formula, so μ_j^3 will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ_j^3 is positive with a negative sign in front of it, the formula will become a convex function. And when μ_j^3 is negative, the formula will become a concave function.

The concavity or convexity of subproblem 2 will lead to different solution approaches.

doi:10.6342/NTU202210101

When μ_j^3 is positive and therefore changing the coefficient of log term into negative, find the point of x_j where the slope is 0, which will be the minimum point of the convex function. If the point of x_j where the slope is 0 falls in the legal range, which is the upper and lower bound of x_j, then it is the solution to x_j. However, if this point does not fall in the legal range of x_j, the upper and lower bound of x_j will be substitute into the objective function to see whichever is smaller, and it will be the solution to x_j.

On the contrary, when μ_j^3 is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to x_j.

The minimum of x_j occurs when both $C_j^a(r_j, m)$ and q_j are at its minimum, which is ϵ. The maximum on the other hand occurs when q_j is 1. We separate x_j from the objective function and derive a subproblem shown as formula (4.40) and constraint shown as formula (4.41).

Table 4.2: Subproblem 2(related to decision variable x_j)

Objective function :

$$Z_{sub2} = \min \sum_{j \in S} \left(\frac{x_j}{m} - \mu_j^3 \log x_j\right) \tag{4.40}$$

Subject to :

$$\epsilon^2 \leq x_j \leq C_j^a(r_j, m) \qquad \forall j \in S \tag{4.41}$$

doi:10.6342/NTU202210101

Algorithm 2: Algorithm for Subproblem 2

```
for each node j do
    Calculate the objective value at x_j = ε^2 denoted as V_1
    Calculate the objective value at x_j = C_j^a(r_j, m) denoted as V_2
    if μ_j^3 ≥ 0 then
        partial differential to x_j
        1/m − μ_j^3 · 1/(x_j ln e) = 0
        x_j = μ_j^3 m
        if x_j is not feasible then
            set x[j] to min(V_1, V_2), which is the boundary value with smaller
              objective value
        end
    else
        set x[j] to min(V_1, V_2), which is the boundary value with smaller
          objective value
    end
end
```

4.2.2.3 Subproblem 3(related to decision variable y_i)

The procedure of dividing y_i from the LR problem is similar with x_i. By extracting all decision variables y_i in the LR problem, we will then obtain subproblem 3. In subproblem 3, μ_i^4 can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 3, we have to take the sign of μ_i^4 into consideration. Linear term will not influence the concavity or convexity of the formula, so μ_i^4 will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ_i^4 is positive with a negative sign in front of it, the formula will become a convex function. And when μ_i^4 is negative, the formula will become a concave function.

The concavity or convexity of subproblem 3 will lead to different solution approaches.

doi:10.6342/NTU202210101

When μ_i^4 is positive and therefore changing the coefficient of log term into negative, find the point of y_i where the slope is 0, which will be the minimum point of the convex function. If the point of y_i where the slope is 0 falls in the legal range, which is the upper and lower bound of y_i, then it is the solution to y_i. However, if this point does not fall in the legal range of y_i, the upper and lower bound of y_i will be substitute into the objective function to see whichever is smaller, and it will be the solution to y_i.

On the contrary, when μ_i^4 is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to y_i.

The minimum of y_i occurs when both $C_i^b(r_i, m)$ and q_i are at its minimum, which is ϵ. The maximum on the other hand occurs when q_i is 1. We separate y_i from the objective function and derive a subproblem shown as formula (4.42) and constraint shown as formula (4.43).

doi:10.6342/NTU202210101

Table 4.3: Subproblem 3(related to decision variable y_i)

Objective function :

$$Z_{sub3} = \min \sum_{i \in S} (-\frac{y_i}{m} - \mu_i^4 \log y_i) \tag{4.42}$$

Subject to :

$$\epsilon^2 \leq y_i \leq C_i^b(r_i, m) \qquad \forall i \in S \tag{4.43}$$

Algorithm 3: Algorithm for Subproblem 3

for *each node i* **do**
  Calculate the objective value at $y_i = \epsilon^2$ denoted as V_1
  Calculate the objective value at $y_i = C_i^b(r_i, m)$ denoted as V_2
  if $\mu_i^4 \geq 0$ **then**
    partial differential to y_i
    $-\frac{1}{m} - \mu_i^4 \frac{1}{y_i \ln e} = 0$
    $y_i = -\mu_i^4 m$
    if y_i *is not feasible* **then**
      set y[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set y[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.2.2.4 Subproblem 4(related to decision variable y_j)

The procedure of dividing y_j from the LR problem is similar with y_j. By extracting all decision variables y_j in the LR problem, we will then obtain subproblem 4. In subproblem 4, μ_j^5 can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 4, we have to take the

doi:10.6342/NTU202210101

sign of μ_j^5 into consideration. Linear term will not influence the concavity or convexity of the formula, so μ_j^5 will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ_j^5 is positive with a negative sign in front of it, the formula will become a convex function. And when μ_j^5 is negative, the formula will become a concave function.

The concavity or convexity of subproblem 4 will lead to different solution approaches. When μ_j^5 is positive and therefore changing the coefficient of log term into negative, find the point of y_j where the slope is 0, which will be the minimum point of the convex function. If the point of y_j where the slope is 0 falls in the legal range, which is the upper and lower bound of y_j, then it is the solution to y_j. However, if this point does not fall in the legal range of y_j, the upper and lower bound of y_j will be substitute into the objective function to see whichever is smaller, and it will be the solution to y_j.

On the contrary, when μ_j^5 is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to y_j.

The minimum of y_j occurs when both $C_j^b(r_j, m)$ and q_j are at its minimum, which is ϵ. The maximum on the other hand occurs when q_j is 1. We separate y_j from the objective function and derive a subproblem shown as formula (4.44) and constraint shown

doi:10.6342/NTU202210101

as formula (4.45).

Table 4.4: Subproblem 4(related to decision variable y_j)

Objective function :
$$Z_{sub4} = \min \sum_{j \in S} (-\frac{y_j}{m} - \mu_j^5 \log y_j) \tag{4.44}$$
Subject to :
$$\epsilon^2 \leq y_j \leq C_j^b(r_j, m) \qquad \forall j \in S \tag{4.45}$$

Algorithm 4: Algorithm for Subproblem 4

---

for *each node j* **do**
  Calculate the objective value at $y_j = \epsilon^2$ denoted as V_1
  Calculate the objective value at $y_j = C_j^b(r_j, m)$ denoted as V_2
  if $\mu_j^5 \geq 0$ **then**
    partial differential to y_j
    $-\frac{1}{m} - \mu_j^5 \frac{1}{y_j \ln e} = 0$
    $y_j = -\mu_j^5 m$
    if y_j *is not feasible* **then**
      set y[j] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set y[j] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

---

doi:10.6342/NTU202210101

4.2.2.5 Subproblem 5(related to decision variable q_i)

By extracting all decision variables q_i in the LR problem, we will then obtain subproblem 5. In subproblem 5, there are 3 multipliers μ_{ij}^1, μ_i^2 and μ_i^4 and all of them can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 5, we have to take the sign of all 4 multipliers μ_{ij}^1, μ_i^2,μ_i^4 and μ_i^6 into consideration. A log function by itself is a concave function, so μ_{ij}^1, μ_i^2,μ_i^4 and μ_i^6 can all play an important role in deciding whether the function is concave or convex.

In this section, we developed a way to easily get the convexity or concavity of this function. We first find the extreme point by finding points with derivative = 0 and will obtain a objective value for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of q_i. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of q_i, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of q_i, if yes then we can return it as our solution to q_i. If the point with derivative = 0 doesn't appears in the legal range of q_i and is considered infeasible, we will return the boundary point of q_i with a smaller objective value as our solution to q_i.

doi:10.6342/NTU202210101

On the contrary, if the objective value with the point of derivative = 0 is bigger than the objective value of the boundary points of q_i, then the function is concave. Therefore, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to q_i.

We separate q_i from the objective function and derive a subproblem shown as formula (4.46) and constraint shown as formula (4.47).

Table 4.5: Subproblem 5(related to decision variable q_i)

Objective function :

$$Z_{sub5} = \min \sum_{i \in S} \sum_{j \in S} (\mu_{ij}^1 \log q_i + \mu_i^2 \log q_i + \mu_i^4 \log q_i + \mu_{ij}^6 \log q_i) \tag{4.46}$$

Subject to :

$$\epsilon \leq q_i \leq 1 \qquad \forall i \in S \tag{4.47}$$

doi:10.6342/NTU202210101

Algorithm 5: Algorithm for Subproblem 5

for *each node i* **do**
  Calculate the objective value at $q_i = \epsilon$ denoted as V_1
  Calculate the objective value at $q_i = 1$ denoted as V_2
  partial differential to q_i
  $\mu_{ij}^1 \frac{1}{q_i \ln e} + \mu_i^2 \frac{1}{q_i \ln e} + \mu_i^4 \frac{1}{q_i \ln e} + \mu_{ij}^6 \frac{1}{q_i \ln e} = 0$
  $\frac{\mu_{ij}^1 + \mu_i^2 + \mu_i^4 + \mu_{ij}^6}{q_i} = 0$
  get solution to q_i
  if $Z_{sub5}(q_i) \leq \min(V_1, V_2)$ **then**
    $Z_{sub5} = convex$
    if q_i *is feasible* **then**
      q[i] = q_i
    else
      set q[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
  else
    set q[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  end
end

4.2.2.6 Subproblem 6(related to decision variable q_j)

By extracting all decision variables q_j in the LR problem, we will then obtain subproblem 6. In subproblem 6, there are 3 multipliers μ_{ij}^1, μ_j^3 and μ_j^5 and all of them can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 6, we have to take the sign of all 3 multipliers μ_{ij}^1, μ_j^3 and μ_j^5 into consideration. A log function by itself is a concave function, so μ_{ij}^1, μ_j^3 and μ_j^5 can all play an important role in deciding whether the function is concave or convex.

To get the convexity or concavity of this function, we first find the extreme point by

doi:10.6342/NTU202210101

finding points with derivative = 0 and will obtain a objective value for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of q_j. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of q_j, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of q_j, if yes then we can return it as our solution to q_j. If the point with derivative = 0 doesn't appears in the legal range of q_j and is considered infeasible, we will return the boundary point of q_j with a smaller objective value as our solution to q_j.

On the contrary, if the objective value with the point of derivative = 0 is bigger than the objective value of the boundary points of q_j, then the function is concave. Therefore, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to q_j.

We separate q_j from the objective function and derive a subproblem shown as formula (4.48) and constraint shown as formula (4.49).

Table 4.6: Subproblem 6(related to decision variable q_j)

Objective function :

$$Z_{sub6} = \min \sum_{i \in S} \sum_{j \in S} (\mu_{ij}^1 \log q_j + \mu_j^3 \log q_j + \mu_j^5 \log q_j) \tag{4.48}$$

Subject to :

$$\epsilon \leq q_j \leq 1 \qquad \forall j \in S \tag{4.49}$$

doi:10.6342/NTU202210101

Algorithm 6: Algorithm for Subproblem 6

for *each node j* **do**
- Calculate the objective value at $q_j = \epsilon$ denoted as V_1
- Calculate the objective value at $q_j = 1$ denoted as V_2
- partial differential to q_j
- $\mu_{ij}^1 \frac{1}{q_j \ln e} + \mu_j^3 \frac{1}{q_j \ln e} + \mu_j^5 \frac{1}{q_j \ln e} = 0$
- $\frac{\mu_{ij}^1 + \mu_j^3 + \mu_j^5}{q_j} = 0$
- get solution to q_j
- **if** $Z_{sub6}(q_j) \leq \min(V_1, V_2)$ **then**
  - $Z_{sub6} = convex$
  - **if** q_j *is feasible* **then**
    - q[j] = q_j
  - **else**
    - set q[j] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  - **end**
- **else**
  - set q[j] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
- **end**

end

4.2.2.7 Subproblem 7(related to decision variable r_i)

By extracting all decision variables r_i in the LR problem, we will then obtain subproblem 7. In subproblem 7, there are 4 multipliers μ_{ij}^1, μ_i^2, $\mu_i^4$and μ_{ij}^6. All of them can be either positive or negative due to relaxing equality constraints,

In order to determine the concavity or convexity of subproblem 7, we have to take the sign of all 4 multipliers μ_{ij}^1, μ_i^2, $\mu_i^4$and μ_{ij}^6 into consideration. A log function by itself is a concave function, soμ_{ij}^1, μ_i^2, $\mu_i^4$and μ_{ij}^6 can all play an important role in deciding whether the function is concave or convex.

To easily and effectively get the convexity or concavity of this function, we first find

doi:10.6342/NTU202210101

the extreme point by finding points with derivative = 0 and will obtain a objective value for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of r_i. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of r_i, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of r_i, if yes then we can return it as our solution to r_i. If the point with derivative = 0 doesn't appears in the legal range of r_i and is considered infeasible, we will return the boundary point of r_i with a smaller objective value as our solution to r_i.

On the contrary, if the objective value with the point of derivative = 0 is bigger than the objective value of the boundary points of r_i, then the function is concave. Therefore, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to r_i.

We separate r_i from the objective function and derive a subproblem shown as formula (4.50) and constraint shown as formula (4.51) and formula (4.52).

doi:10.6342/NTU202210101

Table 4.7: Subproblem 7(related to decision variable r_i)

Objective function :

$$Z_{sub7} = \min \sum_{i \in S} \sum_{j \in S} \Big[\frac{C_i^b(r_i, m)}{m} + \mu_{ij}^1 \log P_{ij}(r_i, m) + \mu_i^2 \log C_i^a(r_i, m) + \mu_i^4 \log C_i^b(r_i, m) + \mu_{ij}^6 C_i^{\tau}(r_i, m) \Big] \tag{4.50}$$

Subject to :

$$\epsilon \le P_{ij}(r_i, m) \le 1 \qquad \forall i \in S, j \in S \tag{4.51}$$

$$r_i \in R_i \qquad \forall i \in S \tag{4.52}$$

Algorithm 7: Algorithm for Subproblem 7

for *each node i* **do**
  Calculate the objective value at $r_i = \max R_i$ denoted as V_1
  Calculate the objective value at $r_i = \min R_i$ denoted as V_2
  Get solution of r_i where Z_{sub7} has the smallest value
  if $Z_{sub7}(r_i, m) \le \min(V_1, V_2)$ **then**
    $Z_{sub7} = convex$
    if r_i *is feasible* **then**
      r[i] = r_i
    else
      set r[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
  else
    set r[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  end
end

4.2.2.8 Subproblem 8(related to decision variable r_j)

By extracting all decision variables r_j in the LR problem, we will then obtain subproblem 8. In subproblem 8, there are 2 multipliers μ_j^3 and μ_j^5. Both of them can be either

doi:10.6342/NTU202210101

positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 8, we have to take both multipliers μ_j^3 and μ_j^5 into consideration. A log function by itself is a concave function, so μ_j^3 and μ_j^5 can both play an important role in deciding whether the function is concave or convex.

To easily and effectively get the convexity or concavity of this function, we first find the extreme point by finding points with derivative = 0 and will obtain a objective value for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of r_j. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of r_j, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of r_j, if yes then we can return it as our solution to r_j. If the point with derivative = 0 doesn't appears in the legal range of r_j and is considered infeasible, we will return the boundary point of r_j with a smaller objective value as our solution to r_j.

On the contrary, if the objective value with the point of derivative = 0 is bigger than the objective value of the boundary points of r_j, then the function is concave. Therefore, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to r_j.

doi:10.6342/NTU202210101

We separate r_j from the objective function and derive a subproblem shown as formula (4.53) and constraint shown as formula (4.54) .

Table 4.8: Subproblem 8(related to decision variable r_j)

Objective function :

$$Z_{sub8} = \min \sum_{j \in S} [\frac{C_j^b(r_j, m)}{m} + \mu_j^3 \log C_j^a(r_j, m) + \mu_j^5 \log C_j^b(r_j, m)] \quad (4.53)$$

Subject to :

$$r_j \in R_j \qquad \forall j \in S \quad (4.54)$$

Algorithm 8: Algorithm for Subproblem 8

for *each node j* **do**
- Calculate the objective value at $r_j = \max R_j$ denoted as V_1
- Calculate the objective value at $r_j = \min R_j$ denoted as V_2
- partial differential to r_j
- $C_j^{b\prime}(r_j, m) + \mu_j^3 \frac{C_j^{a\prime}(r_j,m)}{C_j^a(r_j,m) \ln e} + \mu_j^5 \frac{C_j^{a\prime}(r_j,m)}{C_j^a(r_j,m) \ln e} = 0$
- get solution to r_j
- **if** $Z_{sub8}(r_j, m) \leq \min(V_1, V_2)$ **then**
  - $Z_{sub8} = convex$
  - **if** r_j *is feasible* **then**
    - r[j] = r_j
  - **else**
    - set r[j] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  - **end**
- **else**
  - set r[j] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
- **end**

end

doi:10.6342/NTU202210101

4.2.2.9 Subproblem 9(related to decision variable Z_{ij})

By extracting all decision variables Z_{ij} in the LR problem, we will then obtain subproblem 9. In subproblem 9, there exists multiplier μ^1_{ij}. It can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 9, we have to take multiplier μ^1_{ij} into consideration. A log function by itself is a concave function, so μ^1_{ij} play an important role in deciding whether the function is concave or convex.

We separate Z_{ij} from the objective function and derive a subproblem shown as formula (4.55) and constraint shown as formula (4.56) .

Table 4.9: Subproblem 9(related to decision variable Z_{ij})

Objective function :

$$Z_{sub9} = \min \sum_{i \in S} \sum_{j \in S} (-\mu^1_{ij} \log Z_{ij}) \tag{4.55}$$

Subject to :

$$\max\{\epsilon^3, \frac{1}{\alpha_{ij}}\} \leq Z_{ij} \leq 1 \qquad \forall i \in S, j \in S \tag{4.56}$$

doi:10.6342/NTU202210101

Algorithm 9: Algorithm for Subproblem 9

for *each node i* **do**
  for *each node j* **do**
    Calculate the objective value at $Z_{ij} = \max\{\epsilon^3, \frac{1}{\alpha_{ij}}\}$ denoted as V_1
    Calculate the objective value at $Z_{ij} = 1$ denoted as V_2
    partial differential to Z_{ij}
    $-\mu_{ij}^{1}\frac{1}{Z_{ij}\ln e} = 0$
    get solution to Z_{ij}
    if $Z_{sub9}(Z_{ij}) \leq \min(V_1, V_2)$ **then**
      $Z_{sub9} = convex$
      if Z_{ij} *is feasible* **then**
        Z[i][j] = Z_{ij}
      else
        set Z[i][j] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
      end
    else
      set Z[i][j] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
  end
end

4.2.2.10 Subproblem 10(related to decision variable β_{ij})

By extracting all decision variables β_{ij} in the LR problem, we will then obtain subproblem 10. In subproblem 10, μ_{ij}^{6}can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 10, we have to take the sign of μ_{ij}^{6} into consideration. Linear term will not influence the concavity or convexity of the formula, so μ_{ij}^{6} will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ_{ij}^{6} is positive with a negative sign in front of it, the formula will become a convex function. And when μ_{ij}^{6} is negative, the formula will

doi:10.6342/NTU202210101

become a concave function.

The concavity or convexity of subproblem 10 will lead to different solution approaches. When μ_{ij}^6 is positive and therefore changing the coefficient of log term into negative, find the point of β_{ij} where the slope is 0, which will be the minimum point of the convex function. If the point of β_{ij} where the slope is 0 falls in the legal range, which is the upper and lower bound of β_{ij}, then it is the solution to β_{ij}. However, if this point does not fall in the legal range of β_{ij}, the upper and lower bound of β_{ij} will be substitute into the objective function to see whichever is smaller, and it will be the solution to β_{ij}.

On the contrary, when μ_{ij}^6 is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to β_{ij}.

The minimum of β_{ij} occurs when $C_i^\tau(ri, m)$ is at its minimum. The maximum on the other hand occurs when $C_i^\tau(ri, m)$ is at its maximum. We separate β_{ij} from the objective function and derive a subproblem shown as formula (4.57) and constraint shown as formula (4.58) .

doi:10.6342/NTU202210101

Table 4.10: Subproblem 10(related to decision variable β_{ij})

Objective function :
$$Z_{sub10} = \min \sum_{i \in S} \sum_{j \in S} (\frac{\beta_{ij}}{m} - \mu_{ij}^6 \log \beta_{ij}) \quad (4.57)$$
Subject to :
$$\epsilon \times C_i^\tau(\min R_i, m) \leq \beta_{ij} \leq C_i^\tau(\max R_i, m) \qquad \forall i \in S, j \in S \quad (4.58)$$

Algorithm 10: Algorithm for Subproblem 10

```
for each node i do
    for each node j do
        Calculate the objective value at β_ij = min C_i^τ(ri, m) denoted as V_1
        Calculate the objective value at β_ij = max C_i^τ(ri, m) denoted as V_2
        partial differential to β_ij
        1/m − μ_ij^6 · 1/(β_ij ln e) = 0
        β_ij = μ_ij^6 m
        if Z_sub10(β_ij) ≤ min(V_1, V_2) then
            Z_sub10 = convex
            if Z_ij is feasible then
                Z[i][j] = Z_ij
            else
                set Z[i][j] to min(V_1, V_2), which is the boundary value with
                  smaller objective value
            end
        else
            set Z[i][j] to min(V_1, V_2), which is the boundary value with smaller
              objective value
        end
    end
end
```


doi:10.6342/NTU202210101

4.3 Model 2 : Many-to-One Relationship

4.3.1 Deal with Decision Variables

Constraint (3.18) contains of the product of decision variables $[q_i \times (1-q_i)^{n-1} \times q_\kappa \times P_{i\kappa}(r_i, m)]$. With these decision variables multiplying, two issues will occur. First, the convexity of the function is being destroyed. Second, decision variables will not be able to ”separate” or ”decomposed” when solving it. In order to deal with it,we let

$$q_i \times (1-q_i)^{n-1} \times q_\kappa \times P_{i\kappa}(r_i, m) = Z_{i\kappa} \qquad \forall i \in S, m \in M \tag{4.59}$$

Apply logarithmic operation on both sides and we relax it to our LR problem with Lagrangian multiplier (μ_i^1).

$$\Rightarrow \log q_i + (n-1)\log(1-q_i) + \log q_\kappa + \log P_{i\kappa}(r_i, m) = \log Z_{i\kappa} \quad \forall i \in S, m \in M \tag{4.60}$$

so that we can avoid decision variables from multiplying.

We derived

$$\Rightarrow \epsilon^{n+2} \leq Z_{i\kappa} \leq 1 \qquad \forall i \in S \tag{4.61}$$

from formulation (4.59).

After the transposition in constraint (4.59) with the replacement of $q_i \times (1-q_i)^{n-1} \times q_\kappa \times P_{i\kappa}(r_i, m)$ to $Z_{i\kappa}$, we derive the formulation

doi:10.6342/NTU202210101

$$\frac{1}{Z_{i\kappa}} \leq 1 + \frac{1}{\tau_i}(\overline{T_{ik}} - 1) \qquad \forall i \in S \qquad (4.62)$$

Let

$$\frac{1}{Z_{i\kappa}} \leq 1 + \frac{1}{\tau_i}(\overline{T_{ik}} - 1) = \alpha_{i\kappa} \qquad \forall i \in S \qquad (4.63)$$

and derive

$$\Rightarrow \frac{1}{\alpha_{i\kappa}} \leq Z_{i\kappa} \qquad \forall i \in S \qquad (4.64)$$

which max { $\epsilon^3, \frac{1}{\alpha_{i\kappa}}$ } is the underbound of $Z_{i\kappa}$.

A LR problem is derived as formula (4.65) and we can later keep applying Lagrangian Relaxation:

$$\min \sum_{i \in S} \Bigg[\frac{C_i^a(r_i,m)q_i + C_\kappa^a(r_\kappa,m)q_\kappa + C_i^b(r_i,m)(1-q_i) + C_\kappa^b(r_\kappa,m)(1-q_\kappa) + C_i^\tau(r_i,m)q_i}{m}$$
$$+ \mu_i^1(\log q_i + n\log(1-q_i) + \log q_\kappa + \log P_{i\kappa}(r_i,m) - \log Z_{i\kappa}) \Bigg]$$

(4.65)

where μ_{ij}^1 can be negative or positive.

The objective function is subject to :

$$\epsilon \leq q_i \leq 1 \qquad \forall i \in S \qquad (4.66)$$

$$\epsilon \leq q_\kappa \leq 1 \qquad (4.67)$$

$$\epsilon \leq P_{i\kappa}(r_i,m) \leq 1 \qquad \forall i \in S, m \in M \qquad (4.68)$$

$$\max\{\epsilon^{n+2}, \frac{1}{\alpha_{i\kappa}}\} \leq Z_{i\kappa} \leq 1 \qquad \forall i \in S \qquad (4.69)$$

$$r_i \in R_i \qquad \forall i \in S \qquad (4.70)$$

doi:10.6342/NTU202210101

$$r_\kappa \in R_\kappa \tag{4.71}$$

If $\lambda_{ik} << 1$, which makes $\frac{1}{\lambda_{ik}}$ a great number, we can see this as a M/G/1 system where transmission time is a random variable governed by a geometric distribution.

$$\Rightarrow Prof : \{n \quad retransmission\} = (1 - Z_{i\kappa})^n Z_{i\kappa} \qquad \forall i \in S, m \in M \tag{4.72}$$

Each retransmission will cost us a timeout interval of τ_i, and the last transmission that succeed will cost a time slot. Based on these assumption we can consider queuing delay in variable $\alpha_{i\kappa}$.

If r_i is a continuous variable then $C_i^a(r_i, m)$ will become a decision variable and we might also need to deal with $C_i^a(r_i, m) \times q_i$.

In model one (One-to-One Relationship), we can consider r_i as a fixed number with limited amount of values in different situations. But it won't be a good solution when model one is extended to model two (Many-to-One Relationship) and model three (Network Tree Structure Relationship), so we will try to solve this problem in this section.

In model two (Many-to-One Relationship) and model three (Network Tree Structure Relationship), there are many-to-one relationships with hub and aggregation of data flow. If we consider r_i as a fixed number with limited amount of values as we did in model one (One-to-One Relationship), there occurs a problem called combinatorial explosion. A combinatorial explosion happens when the combinatorics of the problem is affected by the input, constraints, and bounds of the problem which leads to the rapid growth of the

doi:10.6342/NTU202210101

complexity of a problem.

Because of the combinatorial explosion, we can no longer see r_i as a fixed number with limited amount of values, we have to reformulate the objective function to keep decision variables from multiplying. We will expand formula (4.65). But before expanding it, we first let

$$(1 - q_i) = I_i \qquad \forall i \in S \tag{4.73}$$

Where

$$0 \leq I_i \leq 1 - \epsilon \qquad \forall i \in S \tag{4.74}$$

And then we expand formula (4.65) and get formula (4.75)

$$\min \quad \sum_{i \in S} \Bigg[\frac{C_i^a(r_i, m)q_i + C_\kappa^a(r_\kappa, m)q_\kappa + C_i^b(r_i, m)I_i + C_\kappa^b(r_\kappa, m) - C_\kappa^b(r_\kappa, m)q_\kappa + C_i^\tau(r_i, m)q_i}{m}$$
$$+ \sum_{i \in S} \mu_i^1 (\log q_i + n \log I_i + \log q_\kappa + \log P_{i\kappa}(r_i, m) - \log Z_{i\kappa}) \Bigg] \tag{4.75}$$

In order to deal with $C_i^a(r_i, m) \times q_i$, $C_\kappa^a(r_\kappa, m) \times q_\kappa$, $C_i^b(r_i, m) \times I_i$ and $C_\kappa^b(r_\kappa, m) \times q_\kappa$

, we let

$$C_i^a(r_i, m) \times q_i = x_i \qquad \forall i \in S \tag{4.76}$$

$$C_\kappa^a(r_\kappa, m) \times q_\kappa = x_\kappa \tag{4.77}$$

$$C_i^b(r_i, m) \times I_i = y_i \qquad \forall i \in S \tag{4.78}$$

$$C_\kappa^b(r_\kappa, m) \times q_\kappa = y_\kappa \tag{4.79}$$

doi:10.6342/NTU202210101

Apply logarithmic operation on both sides, and we relax these constraints into our LR problem with respective Lagrangian multipliers(μ).

$$\Rightarrow \log C_i^a(r_i, m) + \log q_i = \log x_i \qquad \forall i \in S \tag{4.80}$$

$$\Rightarrow \log C_\kappa^a(r_\kappa, m) + \log q_\kappa = \log x_\kappa \tag{4.81}$$

$$\Rightarrow \log C_i^b(r_i, m) + \log I_i = \log y_i \qquad \forall i \in S \tag{4.82}$$

$$\Rightarrow \log C_\kappa^b(r_\kappa, m) + \log q_\kappa = \log y_\kappa \tag{4.83}$$

A new LR function is derived as the formula below:

$$\begin{aligned}
\min \quad & \sum_{i \in S} \left(\frac{x_i + x_\kappa + y_i + C_\kappa^b(r_\kappa, m) - y_\kappa + C_i^\tau(r_i, m) q_i}{m} \right) \\
& + \sum_{i \in S} \mu_i^1 (\log q_i + (n-1)\log I_i + \log q_\kappa + \log P_{i\kappa}(r_i, m) - \log Z_{i\kappa}) \\
& + \sum_{i \in S} \mu_i^2 (\log C_i^a(r_i, m) + \log q_i - \log x_i) \\
& + \mu^3 (\log C_\kappa^a(r_\kappa, m) + \log q_\kappa - \log x_\kappa) \\
& + \sum_{i \in S} \mu_i^4 (\log C_i^b(r_i, m) + \log I_i - \log y_i) \\
& + \mu^5 (\log C_\kappa^b(r_\kappa, m) + \log q_\kappa - \log y_\kappa)
\end{aligned} \tag{4.84}$$

There still exists decision variables $C_i^\tau(r_i, m)$ and q_i multiplying, and we also have to deal with it.

Let

$$C_i^\tau(r_i, m) q_i = \beta_{ik} \qquad \forall i \in S, m \in M \tag{4.85}$$

doi:10.6342/NTU202210101

Apply logarithmic operation on both sides,

$$\Rightarrow \log C_i^{\tau}(r_i, m) + \log q_i = \log \beta_{i\kappa} \qquad \forall i \in S, m \in M \tag{4.86}$$

Multiply both side with the upper bound and lower bound of $C_i^{\tau}(r_i, m)$ and q_i and we obtain the upper and lower bound of $\beta_{i\kappa}$ as shown as formula(4.87).

$$C_i^{\tau}(\min R_i, m) \times \epsilon \leq \beta_{i\kappa} \leq C_i^{\tau}(\max R_i, m) \qquad \forall i \in S, m \in M \tag{4.87}$$

And because constraint (4.73) contains of 2 decision variables, we also relax it to our LR problem with a multiplier μ^7. The final form of LR problem in model 2(Many-to-One Relationship) is shown as formula (4.88):

$$\begin{aligned}
\min \quad & \sum_{i \in S} \left(\frac{x_i + x_\kappa + y_i + C_\kappa^b(r_\kappa, m) - y_\kappa + \beta_{ik}}{m} \right) \\
& + \sum_{i \in S} \mu_i^1 (\log q_i + (n-1) \log I_i + \log q_\kappa + \log P_{i\kappa}(r_i, m) - \log Z_{i\kappa}) \\
& + \sum_{i \in S} \mu_i^2 (\log C_i^a(r_i, m) + \log q_i - \log x_i) \\
& + \sum_{i \in S} \mu^3 (\log C_\kappa^a(r_\kappa, m) + \log q_\kappa - \log x_\kappa) \\
& + \sum_{i \in S} \mu_i^4 (\log C_i^b(r_i, m) + \log I_i - \log y_i) \\
& + \sum_{i \in S} \mu^5 (\log C_\kappa^b(r_\kappa, m) + \log q_\kappa - \log y_\kappa) \\
& + \sum_{i \in S} \mu_i^6 (\log C_i^{\tau}(r_i, m) + \log q_i - \log \beta_{ik}) \\
& + \sum_{i \in S} \mu_i^7 (I_i + q_i - 1)
\end{aligned} \tag{4.88}$$

doi:10.6342/NTU202210101

The objective function is subject to :

$$\epsilon \le q_i \le 1 \qquad \forall i \in S \tag{4.89}$$

$$0 \le I_i \le 1-\epsilon \qquad \forall i \in S \tag{4.90}$$

$$\epsilon \le q_\kappa \le 1 \tag{4.91}$$

$$\epsilon \le P_{i\kappa}(r_i, m) \le 1 \qquad \forall i \in S, m \in M \tag{4.92}$$

$$\max\{\epsilon^{n+2}, \frac{1}{\alpha_{i\kappa}}\} \le Z_{i\kappa} \le 1 \qquad \forall i \in S \tag{4.93}$$

$$\epsilon^2 \le x_i \le C_i^a(r_i, m) \qquad \forall i \in S \tag{4.94}$$

$$\epsilon^2 \le x_\kappa \le C_\kappa^a(r_\kappa, m) \tag{4.95}$$

$$\epsilon^2 \le y_i \le C_i^b(r_i, m) \qquad \forall i \in S \tag{4.96}$$

$$\epsilon^2 \le y_\kappa \le C_\kappa^b(r_\kappa, m) \tag{4.97}$$

$$C_i^\tau(\min R_i, m) \times \epsilon \le \beta_{i\kappa} \le C_i^\tau(\max R_i, m) \qquad \forall i \in S, m \in M \tag{4.98}$$

$$r_i \in R_i \qquad \forall i \in S \tag{4.99}$$

$$r_\kappa \in R_\kappa \tag{4.100}$$

Multipliers $\mu^1, \mu^2, \mu^3, \mu^4, \mu^5, \mu^6$ and μ^7 respectively represents the vectors of $\{\mu_i^1\}$, $\{\mu_i^2\}$, $\{\mu^3\}$, $\{\mu_i^4\}$, $\{\mu^5\}$, $\{\mu_i^6\}$and , $\{\mu_i^7\}$. They are either positive or negative due to relaxing equality constraints.

doi:10.6342/NTU202210101

4.3.2 The LR Subproblems

To solve this Lagrangian Relaxation problem easily and effectively, we can divide the problem into subproblems and solve them respectively. The way of decomposing this Lagrangian Relaxation problem is by separating the decision variables. Each decision variable will form a subproblem and because the original problem is a minimization problem, the subproblems will also be a minimization problem. Each subproblems will then be dealt with different solution approaches base on their characteristics. This divide and conquer technique will be introduce and implement in this section.

4.3.2.1 Subproblem 1(related to decision variable x_i)

By extracting all decision variables x_i in the LR problem, we will then obtain subproblem 1. In subproblem 1, μ_i^2 can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 1, we have to take the sign of μ_i^2 into consideration. Linear term will not influence the concavity or convexity of the formula, so μ_i^2 will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ_i^2 is positive with a negative sign in front of it, the formula will become a convex function. And when μ_i^2 is negative, the formula will become a concave function.

doi:10.6342/NTU202210101

The concavity or convexity of subproblem 1 will lead to different solution approaches. When μ_i^2 is positive and therefore changing the coefficient of log term into negative, find the point of x_i where the slope is 0, which will be the minimum point of the convex function. If the point of x_i where the slope is 0 falls in the legal range, which is the upper and lower bound of x_i, then it is the solution to x_i. However, if this point does not fall in the legal range of x_i, the upper and lower bound of x_i will be substitute into the objective function to see whichever is smaller, and it will be the solution to x_i.

On the contrary, when μ_i^2 is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to x_i.

The minimum of x_i occurs when both $C_i^a(r_i, m)$ and q_i are at its minimum, which is ϵ. The maximum on the other hand occurs when q_i is 1. We separate x_i from the objective function and derive a subproblem shown as formula (4.101) and constraint shown as formula (4.102).

doi:10.6342/NTU202210101

Table 4.11: Subproblem 1(related to decision variable x_i)

Objective function :

$$Z_{sub1} = \min \sum_{i \in S} (\frac{x_i}{m} - \mu_i^2 \log x_i) \tag{4.101}$$

Subject to :

$$\epsilon^2 \le x_i \le C_i^a(r_i, m) \qquad \forall i \in S \tag{4.102}$$

Algorithm 11: Algorithm for Subproblem 1

for *each node i* **do**
  Calculate the objective value at $x_i = \epsilon^2$ denoted as V_1
  Calculate the objective value at $x_i = C_i^a(r_i, m)$ denoted as V_2
  if $\mu_i^2 \ge 0$ **then**
    partial differential to x_i
    $\frac{1}{m} - \mu_i^2 \frac{1}{x_i \ln e} = 0$
    $x_i = \mu_i^2 m$
    if x_i *is not feasible* **then**
      set x[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set x[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.3.2.2 Subproblem 2(related to decision variable x_κ)

The procedure of dividing x_κ from the LR problem is similar with x_i. By extracting all decision variables x_κ in the LR problem, we will then obtain subproblem 2. In subproblem 2, μ^3 can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 2, we have to take the sign of μ^3 into consideration. Linear term will not influence the concavity or convexity

doi:10.6342/NTU202210101

of the formula, so μ^3 will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ^3 is positive with a negative sign in front of it, the formula will become a convex function. And when μ^3 is negative, the formula will become a concave function.

The concavity or convexity of subproblem 2 will lead to different solution approaches. When μ^3 is positive and therefore changing the coefficient of log term into negative, find the point of x_κ where the slope is 0, which will be the minimum point of the convex function. If the point of x_κ where the slope is 0 falls in the legal range, which is the upper and lower bound of x_κ, then it is the solution to x_κ. However, if this point does not fall in the legal range of x_κ, the upper and lower bound of x_κ will be substitute into the objective function to see whichever is smaller, and it will be the solution to x_κ.

On the contrary, when μ^3 is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to x_κ.

The minimum of x_κ occurs when both $C^a_\kappa(r_\kappa, m)$ and q_κ are at its minimum, which is ϵ. The maximum on the other hand occurs when q_κ is 1. We separate x_κ from the objective function and derive a subproblem shown as formula (4.103) and constraint shown as formula (4.104).

doi:10.6342/NTU202210101

Table 4.12: Subproblem 2(related to decision variable x_κ)

Objective function :

$$Z_{sub2} = \min(\frac{x_\kappa}{m} - \mu^3 \log x_\kappa) \quad (4.103)$$

Subject to :

$$\epsilon^2 \leq x_\kappa \leq C^a_\kappa(r_\kappa, m) \quad (4.104)$$

Algorithm 12: Algorithm for Subproblem 2

Calculate the objective value at $x_\kappa = \epsilon^2$ denoted as V_1
Calculate the objective value at $x_\kappa = C^a_k(r_\kappa, m)$ denoted as V_2
if $\mu^3 \geq 0$ **then**
  partial differential to x_κ
  $\frac{1}{m} - \mu^3 \frac{1}{x_\kappa \ln e} = 0$
  $x_\kappa = \mu^3 m$
  if x_κ *is not feasible* **then**
    set x[k] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
else
  set x[k] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
end

4.3.2.3 Subproblem 3(related to decision variable y_i)

The procedure of dividing y_i from the LR problem is similar with x_i. By extracting all decision variables y_i in the LR problem, we will then obtain subproblem 3. In subproblem 3, μ^4_i can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 3, we have to take the sign of μ^4_i into consideration. Linear term will not influence the concavity or convexity

doi:10.6342/NTU202210101

of the formula, so μ_i^4 will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ_i^4 is positive with a negative sign in front of it, the formula will become a convex function. And when μ_i^4 is negative, the formula will become a concave function.

The concavity or convexity of subproblem 3 will lead to different solution approaches. When μ_i^4 is positive and therefore changing the coefficient of log term into negative, find the point of y_i where the slope is 0, which will be the minimum point of the convex function. If the point of y_i where the slope is 0 falls in the legal range, which is the upper and lower bound of y_i, then it is the solution to y_i. However, if this point does not fall in the legal range of y_i, the upper and lower bound of y_i will be substitute into the objective function to see whichever is smaller, and it will be the solution to y_i.

On the contrary, when μ_i^4 is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to y_i.

The minimum of y_i occurs when both $C_i^b(r_i, m)$ and q_i are at its minimum, which is ϵ. The maximum on the other hand occurs when q_i is 1. We separate y_i from the objective function and derive a subproblem shown as formula (4.105) and constraint shown as formula (4.106).

doi:10.6342/NTU202210101

Table 4.13: Subproblem 3(related to decision variable y_i)

Objective function :
$$Z_{sub3} = \min \sum_{i \in S} (\frac{y_i}{m} - \mu_i^4 \log y_i) \quad (4.105)$$
Subject to :
$$\epsilon^2 \leq y_i \leq C_i^b(r_i, m) \qquad \forall i \in S \quad (4.106)$$

Algorithm 13: Algorithm for Subproblem 3

for *each node i* **do**
  Calculate the objective value at $y_i = \epsilon^2$ denoted as V_1
  Calculate the objective value at $y_i = C_i^b(r_i, m)$ denoted as V_2
  if $\mu_i^4 \geq 0$ **then**
    partial differential to y_i
    $\frac{1}{m} - \mu_i^4 \frac{1}{y_i \ln e} = 0$
    $y_i = \mu_i^4 m$
    if y_i *is not feasible* **then**
      set y[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set y[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.3.2.4 Subproblem 4(related to decision variable y_κ)

The procedure of dividing y_κ from the LR problem is similar with y_κ. By extracting all decision variables y_κ in the LR problem, we will then obtain subproblem 4. In subproblem 4, μ^5 can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 4, we have to take the sign of μ^5 into consideration. Linear term will not influence the concavity or convexity

doi:10.6342/NTU202210101

of the formula, so μ^5 will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ^5 is positive with a negative sign in front of it, the formula will become a convex function. And when μ^5 is negative, the formula will become a concave function.

The concavity or convexity of subproblem 4 will lead to different solution approaches. When μ^5 is positive and therefore changing the coefficient of log term into negative, find the point of y_κ where the slope is 0, which will be the minimum point of the convex function. If the point of y_κ where the slope is 0 falls in the legal range, which is the upper and lower bound of y_κ, then it is the solution to y_κ. However, if this point does not fall in the legal range of y_κ, the upper and lower bound of y_κ will be substitute into the objective function to see whichever is smaller, and it will be the solution to y_κ.

On the contrary, when μ^5 is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to y_κ.

The minimum of y_κ occurs when both $C^b_\kappa(r_\kappa, m)$ and q_κ are at its minimum, which is ϵ. The maximum on the other hand occurs when q_κ is 1. We separate y_κ from the objective function and derive a subproblem shown as formula (4.107) and constraint shown as formula (4.108).

doi:10.6342/NTU202210101

Table 4.14: Subproblem 4(related to decision variable y_κ)

Objective function :

$$Z_{sub4} = \min(-\frac{y_\kappa}{m} - \mu^5 \log y_\kappa) \tag{4.107}$$

Subject to :

$$\epsilon^2 \leq y_\kappa \leq C_\kappa^b(r_\kappa, m) \tag{4.108}$$

Algorithm 14: Algorithm for Subproblem 4

Calculate the objective value at $y_\kappa = \epsilon^2$ denoted as V_1
Calculate the objective value at $y_\kappa = C_k^b(r_\kappa, m)$ denoted as V_2
if $\mu^5 \geq 0$ **then**
  partial differential to y_κ
  $-\frac{1}{m} - \mu^5 \frac{1}{y_\kappa \ln e} = 0$
  $y_\kappa = -\mu^5 m$
  if y_κ *is not feasible* **then**
    set y[k] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
else
  set y[k] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
end

4.3.2.5 Subproblem 5(related to decision variable q_i)

By extracting all decision variables q_i in the LR problem, we will then obtain subproblem 5. In subproblem 5, there are 4 multipliers μ_i^1,μ_i^2,μ_i^6 and μ_i^7 and all of them can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 5, we have to take the sign of μ_i^1 ,$\mu_i^2$and μ_i^6into consideration, but not μ_i^7 because q_i is linear so it won't affect

doi:10.6342/NTU202210101

the concavity or convexity of subproblem 5. A log function by itself is a concave function, so μ_i^1 ,$\mu_i^2$and μ_i^6 can all play an important role in deciding whether the function is concave or convex.

In this section, we developed a way to easily get the convexity or concavity of this function. We first find the extreme point by finding points with derivative = 0 and will obtain a objective value for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of q_i. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of q_i, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of q_i, if yes then we can return it as our solution to q_i. If the point with derivative = 0 doesn't appears in the legal range of q_i and is considered infeasible, we will return the boundary point of q_i with a smaller objective value as our solution to q_i.

On the contrary, if the objective value with the point of derivative = 0 is bigger than the objective value of the boundary points of q_i, then the function is concave. Therefore, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to q_i.

We separate q_i from the objective function and derive a subproblem shown as formula (4.109) and constraint shown as formula (4.110).

doi:10.6342/NTU202210101

Table 4.15: Subproblem 5(related to decision variable q_i)

Objective function :

$$Z_{sub5} = \min \sum_{i \in S} (\mu_i^1 \log q_i + \mu_i^2 \log q_i + \mu^6 \log q_i + \mu_i^7 q_i) \tag{4.109}$$

Subject to :

$$\epsilon \le q_i \le 1 \qquad \forall i \in S \tag{4.110}$$

Algorithm 15: Algorithm for Subproblem 5

for *each node i* **do**
- Calculate the objective value at $q_i = \epsilon$ denoted as V_1
- Calculate the objective value at $q_i = 1$ denoted as V_2
- partial differential to q_i
- $\mu_i^1 \frac{1}{q_i \ln e} + \mu_i^2 \frac{1}{q_i \ln e} + \mu_i^6 \frac{1}{q_i \ln e} + \mu_i^7 = 0$
- $\frac{\mu_i^1 + \mu_i^2 + \mu_i^6}{q_i} = -\mu_i^7$
- $q_i = -\frac{\mu_i^1 + \mu_i^2 + \mu_i^6}{\mu_i^7}$
- **if** $Z_{sub5}(q_i) \le \min(V_1, V_2)$ **then**
  - Z$_{sub5} = convex$
  - **if** q_i *is feasible* **then**
    - q[i] = q_i
  - **else**
    - set q[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  - **end**
- **else**
  - set q[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
- **end**

end

4.3.2.6 Subproblem 6(related to decision variable q_κ)

By extracting all decision variables q_κ in the LR problem, we will then obtain subproblem 6. In subproblem 6, there are 3 multipliers μ_i^1, μ_k^3 and μ_k^5 and all of them can be either positive or negative due to relaxing equality constraints.

doi:10.6342/NTU202210101

In order to determine the concavity or convexity of subproblem 6, we have to take the sign of all 3 multipliers μ_i^1, μ_k^3 and μ_k^5 into consideration. A log function by itself is a concave function, so μ_i^1, μ_k^3 and μ_k^5 can all play an important role in deciding whether the function is concave or convex.

To get the convexity or concavity of this function, we first find the extreme point by finding points with derivative = 0 and will obtain a objective value for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of q_κ. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of q_κ, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of q_κ, if yes then we can return it as our solution to q_κ. If the point with derivative = 0 doesn't appears in the legal range of q_κ and is considered infeasible, we will return the boundary point of q_κ with a smaller objective value as our solution to q_κ.

On the contrary, if the objective value with the point of derivative = 0 is bigger than the objective value of the boundary points of q_κ, then the function is concave. Therefore, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to q_κ.

We separate q_κ from the objective function and derive a subproblem shown as formula (4.111) and constraint shown as formula (4.112).

doi:10.6342/NTU202210101

Table 4.16: Subproblem 6(related to decision variable q_κ)

Objective function :

$$Z_{sub6} = \min \sum_{i \in S} (\mu_i^1 \log q_\kappa + \mu_k^3 \log q_\kappa + \mu_k^5 \log q_\kappa) \quad (4.111)$$

Subject to :

$$\epsilon \leq q_\kappa \leq 1 \quad (4.112)$$

Algorithm 16: Algorithm for Subproblem 6

Calculate the objective value at $q_\kappa = \epsilon$ denoted as V_1
Calculate the objective value at $q_\kappa = 1$ denoted as V_2
partial differential to q_κ
$\mu_i^1 \frac{1}{q_\kappa \ln e} + \mu_k^3 \frac{1}{q_\kappa \ln e} + \mu_k^5 \frac{1}{q_\kappa \ln e} = 0$
$\frac{\mu_i^1 + \mu_k^3 + \mu_k^5}{q_\kappa} = 0$
get solution to q_κ
if $Z_{sub6}(q_\kappa) \leq \min(V_1, V_2)$ **then**
  $Z_{sub6} = convex$
  if q_κ *is feasible* **then**
    qk = q_κ
  else
    set qk to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
else
  set qk to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
end

4.3.2.7 Subproblem 7(related to decision variable r_i)

By extracting all decision variables r_i in the LR problem, we will then obtain subproblem 7. In subproblem 7, there are 4 multipliers μ_i^1, μ_i^2, μ_i^4 and μ_i^6. All of them can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 7, we have to take

doi:10.6342/NTU202210101

the sign of all 4 multipliers μ_i^1, μ_i^2, μ_i^4 and μ_i^6 into consideration. A log function by itself is a concave function, so μ_i^1, μ_i^2, μ_i^4 and μ_i^6 can all play an important role in deciding whether the function is concave or convex.

To easily and effectively get the convexity or concavity of this function, we first find the extreme point by finding points with derivative = 0 and will obtain a objective value for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of r_i. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of r_i, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of r_i, if yes then we can return it as our solution to r_i. If the point with derivative = 0 doesn't appears in the legal range of r_i and is considered infeasible, we will return the boundary point of r_i with a smaller objective value as our solution to r_i.

On the contrary, if the objective value with the point of derivative = 0 is bigger than the objective value of the boundary points of r_i, then the function is concave. Therefore, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to r_i.

We separate r_i from the objective function and derive a subproblem shown as formula (4.113) and constraint shown as formula (4.114) and formula (4.115).

doi:10.6342/NTU202210101

Table 4.17: Subproblem 7(related to decision variable r_i)

Objective function :

$$Z_{sub7} = \min \sum_{i \in S} \left[\mu_i^1 \log P_{i\kappa}(r_i, m) + \mu_i^2 \log C_i^a(r_i, m) + \mu_i^4 \log C_i^b(r_i, m) + \mu_i^6 C_i^\tau(r_i, m) \right] \tag{4.113}$$

Subject to :

$$\epsilon \le P_{i\kappa}(r_i, m) \le 1 \qquad \forall i \in S \tag{4.114}$$

$$r_i \in R_i \qquad \forall i \in S \tag{4.115}$$

Algorithm 17: Algorithm for Subproblem 7

---

for *each node i* **do**
  Calculate the objective value at $r_i = \max R_i$ denoted as V_1
  Calculate the objective value at $r_i = \min R_i$ denoted as V_2
  Get solution of r_i where Z_{sub7} has the smallest value
  if $Z_{sub7}(r_i, m) \le \min(V_1, V_2)$ **then**
    $Z_{sub7} = convex$
    if r_i *is feasible* **then**
      r[i] = r_i
    else
      set r[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set r[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.3.2.8 Subproblem 8(related to decision variable r_κ)

By extracting all decision variables r_κ in the LR problem, we will then obtain subproblem 8. In subproblem 8, there are 2 multipliers μ_k^3 and μ_k^5. Both of them can be either positive or negative due to relaxing equality constraints.

doi:10.6342/NTU202210101

In order to determine the concavity or convexity of subproblem 8, we have to take both multipliers μ_k^3 and μ_k^5 into consideration. A log function by itself is a concave function, so μ_k^3 and μ_k^5 can both play an important role in deciding whether the function is concave or convex.

To easily and effectively get the convexity or concavity of this function, we first find the extreme point by finding points with derivative = 0 and will obtain a objective value for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of r_κ. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of r_κ, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of r_κ, if yes then we can return it as our solution to r_κ. If the point with derivative = 0 doesn't appears in the legal range of r_κ and is considered infeasible, we will return the boundary point of r_κ with a smaller objective value as our solution to r_κ.

On the contrary, if the objective value with the point of derivative = 0 is bigger than the objective value of the boundary points of r_κ, then the function is concave. Therefore, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to r_κ.

We separate r_j from the objective function and derive a subproblem shown as formula

doi:10.6342/NTU202210101

(4.116) and constraint shown as formula (4.117) .

Table 4.18: Subproblem 8(related to decision variable r_κ)

Objective function :

$$Z_{sub8} = \min \quad \frac{C_k^b(r_\kappa, m)}{m} + \mu_k^3 \log C_k^a(r_\kappa, m) + \mu_k^5 \log C_k^b(r_\kappa, m) \tag{4.116}$$

Subject to :

$$r_\kappa \in R_\kappa \tag{4.117}$$

Algorithm 18: Algorithm for Subproblem 8

Calculate the objective value at $r_\kappa = \max R_\kappa$ denoted as V_1
Calculate the objective value at $r_\kappa = \min R_\kappa$ denoted as V_1
partial differential to r_κ
$\frac{C_k^{b\prime}(r_\kappa,m)}{m} + \mu_k^3 \frac{C_k^{a\prime}(r_\kappa,m)}{C_k^a(r_\kappa,m)\ln e} + \mu_k^5 \frac{C_k^{a\prime}(r_\kappa,m)}{C_k^a(r_\kappa,m)\ln e} = 0$
get solution to r_κ
if $Z_{sub8}(r_\kappa, m) \leq \min(V_1, V_2)$ **then**
    Z$_{sub8} = convex$
    if r_κ *is feasible* **then**
        rk = r_κ
    else
        set rk to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
else
    set rk to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
end

doi:10.6342/NTU202210101

4.3.2.9 Subproblem 9(related to decision variable $Z_{i\kappa}$)

By extracting all decision variables $Z_{i\kappa}$ in the LR problem, we will then obtain subproblem 9. In subproblem 9, there exists multiplier μ_i^1. It can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 9, we have to take multiplier μ_i^1 into consideration. A log function by itself is a concave function, so μ_i^1 can both an important role in deciding whether the function is concave or convex.

We separate $Z_{i\kappa}$ from the objective function and derive a subproblem shown as formula (4.118) and constraint shown as formula (4.119) .

Table 4.19: Subproblem 9(related to decision variable $Z_{i\kappa}$)

Objective function :

$$Z_{sub9} = \min \sum_{i \in S} (-\mu_i^1 \log Z_{i\kappa}) \tag{4.118}$$

Subject to :

$$\max\{\epsilon^{n+2}, \frac{1}{\alpha_{i\kappa}}\} \le Z_{i\kappa} \le 1 \qquad \forall i \in S \tag{4.119}$$

doi:10.6342/NTU202210101

Algorithm 19: Algorithm for Subproblem 9

```
for each node i do
    Calculate the objective value at Z_{iκ} = max{ϵ^{n+2}, 1/α_{iκ}} denoted as V_1
    Calculate the objective value at Z_{iκ} = 1 denoted as V_2
    partial differential to Z_{iκ}
    −μ_i^1 (1 / (Z_{iκ} ln e)) = 0
    get solution to Z_{iκ}
    if Z_{sub9}(Z_{iκ}) ≤ min(V_1, V_2) then
        Z_{sub9} = convex
        if Z_{iκ} is feasible then
            Z[i] = Z_{iκ}
        else
            set Z[i] to min(V_1, V_2), which is the boundary value with smaller
              objective value
        end
    else
        set Z[i] to min(V_1, V_2), which is the boundary value with smaller
          objective value
    end
end
```

4.3.2.10 Subproblem 10(related to decision variable $\beta_{i\kappa}$)

The procedure of dividing $\beta_{i\kappa}$ from the LR problem is similar with x_i. By extracting all decision variables $\beta_{i\kappa}$ in the LR problem, we will then obtain subproblem 10. In subproblem 10, μ_i^6 can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 3, we have to take the sign of μ_i^6 into consideration. Linear term will not influence the concavity or convexity of the formula, so μ_i^6 will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ_i^6 is positive with a negative sign in front of it, the formula will become a convex function. And when μ_i^6 is negative, the formula will become a concave function.

doi:10.6342/NTU202210101

The concavity or convexity of subproblem 10 will lead to different solution approaches. When μ_i^6 is positive and therefore changing the coefficient of log term into negative, find the point of $\beta_{i\kappa}$ where the slope is 0, which will be the minimum point of the convex function. If the point of $\beta_{i\kappa}$ where the slope is 0 falls in the legal range, which is the upper and lower bound of $\beta_{i\kappa}$, then it is the solution to $\beta_{i\kappa}$. However, if this point does not fall in the legal range of $\beta_{i\kappa}$, the upper and lower bound of $\beta_{i\kappa}$ will be substitute into the objective function to see whichever is smaller, and it will be the solution to $\beta_{i\kappa}$.

On the contrary, when μ_i^6 is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to $\beta_{i\kappa}$.

The minimum of $\beta_{i\kappa}$ occurs when both all $C_i^\tau(r_i, m)$,$\frac{1}{Z_{i\kappa}}$ and q_i are at its minimum, which is $C_i^\tau(\min R_i, m) \times \epsilon$. The maximum on the other hand occurs when $C_i^\tau(r_i, m)$,$\frac{1}{Z_{i\kappa}}$ and q_i are at its maximum. We separate $\beta_{i\kappa}$ from the objective function and derive a subproblem shown as formula (4.120) and constraint shown as formula (4.121).

doi:10.6342/NTU202210101

Table 4.20: Subproblem 10(related to decision variable $\beta_{i\kappa}$)

Objective function : $$Z_{sub10} = \min \sum_{i \in S} (\frac{\beta_{i\kappa}}{m} - \mu_i^6 \log \beta_{i\kappa}) \quad (4.120)$$ **Subject to :** $$C_i^\tau(\min R_i, m) \times \epsilon \le \beta_{i\kappa} \le C_i^\tau(\max R_i, m) \qquad \forall i \in S, m \in M \quad (4.121)$$

Algorithm 20: Algorithm for Subproblem 10

for *each node i* **do**
  Calculate the objective value at $\beta_{i\kappa} = C_i^\tau(\min R_i, m) \times \epsilon$ denoted as V_1
  Calculate the objective value at $\beta_{i\kappa} = C_i^\tau(\max R_i, m)$ denoted as V_2
  if $\mu_i^6 \ge 0$ **then**
    partial differential to $\beta_{i\kappa}$
    $\frac{1}{m} - \mu_i^6 \frac{1}{\beta_{i\kappa} \ln e} = 0$
    $\beta_{i\kappa} = \mu_i^6 m$
    if $\beta_{i\kappa}$ *is not feasible* **then**
      set β*[i] to* $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set β*[i] to* $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.3.2.11 Subproblem 11(related to decision variable I_i)

By extracting all decision variables I_i in the LR problem, we will then obtain subproblem 11. In subproblem 11, there are 2 multipliers μ_i^1 and μ_i^4. μ_i^1 and μ_i^4 can both be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 11, we have to take both multipliers μ_i^1 and μ_i^4 into consideration. Linear term will not influence the concavity

doi:10.6342/NTU202210101

or convexity of the formula so we don't have to look at μ_i^7. A log function by itself is a concave function, so μ_i^1 and μ_i^4 will both play an important role in deciding whether it is concave or convex.

To easily and effectively get the convexity or concavity of this function, we first find the extreme point by finding points with derivative = 0 and will obtain a objective value for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of I_i. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of I_i, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of I_i, if yes then we can return it as our solution to I_i. If the point with derivative = 0 doesn't appears in the legal range of I_i and is considered infeasible, we will return the boundary point of I_i with a smaller objective value as our solution to I_i.

The minimum of I_i occurs when q_i is at its maximum, which is 0. The maximum on the other hand occurs when q_i is at its minimum, which is 1. We separate I_i from the objective function and derive a subproblem shown as formula (4.122) and constraint shown as formula (4.123) .

doi:10.6342/NTU202210101

Table 4.21: Subproblem 10(related to decision variable I_i)

Objective function :

$$Z_{sub11} = \min \sum_{i \in S} (\mu_i^1 (n-1) \log I_i + \mu_i^4 \log I_i + \mu_i^7 I_i) \tag{4.122}$$

Subject to :

$$0 \le I_i \le 1 - \epsilon \qquad \forall i \in S \tag{4.123}$$

Algorithm 21: Algorithm for Subproblem 11

for *each node i* **do**
  Calculate the objective value at $I_i = 0$ denoted as V_1
  Calculate the objective value at $I_i = 1 - \epsilon$ denoted as V_2
  partial differential to I_i
  $\mu_i^1 \frac{n-1}{I_i \ln e} + \mu_i^4 \frac{1}{I_i \ln e} + \mu_i^7 = 0$
  $I_i = -\frac{(n-1)\mu_i^1 + \mu_i^4}{\mu_i^7}$
  if $Z_{sub11}(I_i) \le \min(V_1, V_2)$ **then**
    $Z_{sub11} = convex$
    if I_i *is feasible* **then**
      I[i] = I_i
    else
      set I[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
  else
    set I[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  end
end

4.4 Model 3 : Network Tree Structure Relationship

4.4.1 Deal with Decision Variables

Constraint (3.31) contains of the product of decision variables. We first take $q_{\theta_i}(1 - q_{\theta_i})^{(v-1)} q_{\kappa R_i} P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ and $q_{\kappa S_i}(1 - q_{\kappa S_i})^{(n-1)} q_\xi P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ into consideration. With

doi:10.6342/NTU202210101

these decision variables multiplying, two issues will occur. First, the convexity of the function is being destroyed . Second, decision variables will not be able to ”separate” or ”decomposed” when solving it. In order to deal with it,we let

$$q_{\theta_i}(1-q_{\theta_i})^{(v-1)}q_{\kappa R_i}P_{\theta_i,\kappa_i}(r_{\theta_i},m)=Z_{\theta_i,\kappa_i} \qquad \forall i\in N, m\in M \tag{4.124}$$

$$q_{\kappa S_i}(1-q_{\kappa S_i})^{(n-1)}q_{\xi}P_{\kappa_i,\xi}(r_{\kappa_i},m)=Z_{\kappa_i,\xi} \qquad \forall i\in N, m\in M \tag{4.125}$$

Apply logarithmic operation on both sides and we relax them to our LR problem with Lagrangian multiplier (μ_i^1 and μ_i^2).

$$\Rightarrow \log q_{\theta_i}+(v-1)\log(1-q_{\theta_i})+\log q_{\kappa R_i}+\log P_{\theta_i,\kappa_i}(r_{\theta_i},m)=\log Z_{\theta_i,\kappa_i} \quad \forall i\in N, m\in M \tag{4.126}$$

$$\Rightarrow \log q_{\kappa S_i}+(n-1)\log(1-q_{\kappa S_i})+\log q_{\xi}+\log P_{\kappa_i,\xi}(r_{\kappa_i},m)=\log Z_{\kappa_i,\xi} \quad \forall i\in N, m\in M \tag{4.127}$$

so that we can avoid decision variables from multiplying.

We derived

$$\Rightarrow \epsilon^{v+2}\le Z_{\theta_i,\kappa_i}\le 1 \qquad \forall i\in N \tag{4.128}$$

from formulation (4.124).

We also derived

$$\Rightarrow \epsilon^{n+2}\le Z_{\kappa_i,\xi}\le 1 \qquad \forall i\in N \tag{4.129}$$

from formulation (4.125).

Before we deal with other decision variables multiplying, we first let

doi:10.6342/NTU202210101

$$(1-q_{\theta_i}) = I_{\theta_i} \qquad \forall i \in N \tag{4.130}$$

$$(1-q_{\kappa R_i}) = I_{\kappa R_i} \qquad \forall i \in N \tag{4.131}$$

$$(1-q_{\kappa S_i}) = I_{\kappa S_i} \qquad \forall i \in N \tag{4.132}$$

Where

$$0 \leq I_{\theta_i} \leq 1-\epsilon \qquad \forall i \in N \tag{4.133}$$

$$0 \leq I_{\kappa R_i} \leq 1-\epsilon \qquad \forall i \in N \tag{4.134}$$

$$0 \leq I_{\kappa S_i} \leq 1-\epsilon \qquad \forall i \in N \tag{4.135}$$

In order to deal with $C^a_{\theta_i}(r_{\theta_i}, m) \times q_{\theta_i}, C^a_{\kappa_i}(r_{\kappa_i}, m) \times q_{\kappa R_i}, C^a_{\xi}(r_{\xi}, m) \times q_{\xi}, C^b_{\theta_i}(r_{\theta_i}, m) \times I_{\theta_i}, C^b_{\kappa_i}(r_{\kappa_i}, m) \times I_{\kappa_i}$ and $C^b_{\xi}(r_{\xi}, m) \times q_{\xi}$, we let

$$C^a_{\theta_i}(r_{\theta_i}, m) \times q_{\theta_i} = x_{\theta_i} \qquad \forall i \in N, m \in M \tag{4.136}$$

$$C^a_{\kappa_i}(r_{\kappa_i}, m) \times q_{\kappa R_i} = x_{\kappa_i} \qquad \forall i \in N, m \in M \tag{4.137}$$

$$C^a_{\xi}(r_{\xi}, m) \times q_{\xi} = x_{\xi} \qquad \forall m \in M \tag{4.138}$$

$$C^b_{\theta_i}(r_{\theta_i}, m) \times I_{\theta_i} = y_{\theta_i} \qquad \forall i \in N, m \in M \tag{4.139}$$

$$C^b_{\kappa_i}(r_{\kappa_i}, m) \times I_{\kappa R_i} = y_{\kappa_i} \qquad \forall i \in N, m \in M \tag{4.140}$$

$$C^b_{\xi}(r_{\xi}, m) \times q_{\xi} = y_{\xi} \qquad \forall m \in M \tag{4.141}$$

Where

doi:10.6342/NTU202210101

$$\epsilon^2 \le x_{\theta_i} \le C^a_{\theta_i}(r_{\theta_i}, m) \qquad \forall i \in N, m \in M \tag{4.142}$$

$$\epsilon^2 \le x_{\kappa_i} \le C^a_{\kappa_i}(r_{\kappa_i}, m) \qquad \forall i \in N, m \in M \tag{4.143}$$

$$\epsilon^2 \le x_{\xi} \le C^a_{\xi}(r_{\xi}, m) \qquad \forall m \in M \tag{4.144}$$

$$\epsilon^2 \le y_{\theta_i} \le C^b_{\theta_i}(r_{\theta_i}, m) \qquad \forall i \in N, m \in M \tag{4.145}$$

$$\epsilon^2 \le y_{\kappa_i} \le C^b_{\kappa_i}(r_{\kappa_i}, m) \qquad \forall i \in N, m \in M \tag{4.146}$$

$$\epsilon^2 \le y_{\xi} \le C^b_{\xi}(r_{\xi}, m) \qquad \forall m \in M \tag{4.147}$$

Apply logarithmic operation on both sides, and we relax these constraints into our LR problem with respective Lagrangian multipliers(μ).

$$\Rightarrow \log C^a_{\theta_i}(r_{\theta_i}, m) + \log q_{\theta_i} = \log x_{\theta_i} \qquad \forall i \in N, m \in M \tag{4.148}$$

$$\Rightarrow \log C^a_{\kappa_i}(r_{\kappa_i}, m) + \log q_{\kappa R_i} = \log x_{\kappa_i} \qquad \forall i \in N, m \in M \tag{4.149}$$

$$\Rightarrow \log C^a_{\xi}(r_{\xi}, m) + \log q_{\xi} = \log x_{\xi} \qquad \forall m \in M \tag{4.150}$$

$$\Rightarrow \log C^b_{\theta_i}(r_{\theta_i}, m) + \log I_{\theta_i} = \log y_{\theta_i} \qquad \forall i \in N, m \in M \tag{4.151}$$

$$\Rightarrow \log C^b_{\kappa_i}(r_{\kappa_i}, m) + \log I_{\kappa R_i} = \log y_{\kappa_i} \qquad \forall i \in N, m \in M \tag{4.152}$$

$$\Rightarrow \log C^b_{\xi}(r_{\xi}, m) + \log q_{\xi} = \log y_{\xi} \qquad \forall m \in M \tag{4.153}$$

A new LR function is derived as the formula below:

doi:10.6342/NTU202210101

$$\min \quad \sum_{i \in N} \Bigg(\frac{x_{\theta_i} \times v + y_{\theta_i} \times v + C^{\tau}_{\theta_i}(r_{\theta_i}, m) q_{\theta_i} \times v + x_{\kappa_i} + y_{\kappa_i} + C^{\tau}_{\kappa_i}(r_{\kappa_i}, m) q_{\kappa S_i} + x_\xi + C^b_\xi(r_\xi, m) - y_\xi}{m}$$

$$+ \sum_{i \in N} \mu_i^1 (\log q_{\theta_i} + (v-1)\log I_{\theta_i} + \log q_{\kappa R_i} + \log P_{\theta_i,\kappa_i}(r_{\theta_i}, m) - \log Z_{\theta_i,\kappa_i})$$

$$+ \sum_{i \in N} \mu_i^2 (\log q_{\kappa S_i} + (n-1)\log I_{\kappa S_i} + \log q_\xi + \log P_{\kappa_i,\xi}(r_{\kappa_i}, m) - \log Z_{\kappa_i,\xi})$$

$$+ \sum_{i \in N} \mu_i^3 (\log C^a_{\theta_i}(r_{\theta_i}, m) + \log q_{\theta_i} - \log x_{\theta_i})$$

$$+ \sum_{i \in N} \mu_i^4 (\log C^a_{\kappa_i}(r_{\kappa_i}, m) + \log q_{\kappa_i} - \log x_{\kappa_i})$$

$$+ \sum_{i \in N} \mu^5 (\log C^a_\xi(r_\xi, m) + \log q_\xi - \log x_\xi)$$

$$+ \sum_{i \in N} \mu_i^6 (\log C^b_{\theta_i}(r_{\theta_i}, m) + \log I_{\theta_i} - \log y_{\theta_i})$$

$$+ \sum_{i \in N} \mu_i^7 (\log C^b_{\kappa_i}(r_{\kappa_i}, m) + \log I_{\kappa R_i} - \log y_{\kappa_i})$$

$$+ \sum_{i \in N} \mu^8 (\log C^b_\xi(r_\xi, m) + \log q_\xi - \log y_\xi)$$

(4.154)

There still exists $C^{\tau}_{\theta_i}(r_{\theta_i}, m) q_{\theta_i}$ and $C^{\tau}_{\kappa_i}(r_{\kappa_i}, m) q_{\kappa S_i}$ which contains decision variables multiplying, so we also have to deal with it.

Let

$$C^{\tau}_{\theta_i}(r_{\theta_i}, m) q_{\theta_i} = \beta_{\theta_i,\kappa_i} \qquad \forall i \in N, m \in M \qquad (4.155)$$

$$C^{\tau}_{\kappa_i}(r_{\kappa_i}, m) q_{\kappa S_i} = \beta_{\kappa_i,\xi} \qquad \forall i \in N, m \in M \qquad (4.156)$$

Apply logarithmic operation on both sides,

$$\Rightarrow \log C^{\tau}_{\theta_i}(r_{\theta_i}, m) + \log q_{\theta_i} = \log \beta_{\theta_i,\kappa_i} \qquad \forall i \in N, m \in M \qquad (4.157)$$

doi:10.6342/NTU202210101

$$\Rightarrow \log C^{\tau}_{\kappa_i}(r_{\kappa_i}, m) + \log q_{\kappa S_i} = \log \beta_{\kappa_i,\xi} \qquad \forall i \in N, m \in M \tag{4.158}$$

Multiply both side with the upper bound and lower bound of $C^{\tau}_{\theta_i}(r_{\theta_i}, m), C^{\tau}_{\kappa_i}(r_{\kappa_i}, m)$ and $q_{\theta_i}, q_{\kappa S_i}$ and we obtain the upper and lower bound of $\beta_{\theta_i,\kappa_i}$ and $\beta_{\kappa_i,\xi}$ as shown as formula(4.159) and (4.160).

$$C^{\tau}_{\theta_i}(\min R_{\theta_i}, m) \times \epsilon \leq \beta_{\theta_i,\kappa_i} \leq C^{\tau}_{\theta_i}(\max R_{\theta_i}, m) \qquad \forall i \in N, m \in M \tag{4.159}$$

$$C^{\tau}_{\kappa_i}(\min R_{\kappa_i}, m) \times \epsilon \leq \beta_{\kappa_i,\xi} \leq C^{\tau}_{\kappa_i}(\max R_{\kappa_i}, m) \qquad \forall i \in N, m \in M \tag{4.160}$$

Now that we've dealt with the products of decision variables in the objective function, we also have to deal with the one in constraint (3.31) . We replace $q_{\theta_i}(1-q_{\theta_i})^{(v-1)}q_{\kappa R_i}P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ with Z_{θ_i,κ_i} and $q_{\kappa S_i}(1-q_{\kappa S_i})^{(n-1)}q_{\xi}P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ with $Z_{\kappa_i,\xi}$ in constraint (3.31) as shown as formula(4.161)

$$\tau_{\theta_i}\left[\frac{1}{Z_{\theta_i,\kappa_i}} - 1\right] + \tau_{\kappa_i}\left[\frac{1}{Z_{\kappa_i,\xi}} - 1\right] + 2 \leq \overline{T_{\theta_i,\xi}} \qquad \forall i \in N, m \in M \tag{4.161}$$

$$\Rightarrow \frac{\tau_{\theta_i}}{Z_{\theta_i,\kappa_i}} - \tau_{\theta_i} + \frac{\tau_{\kappa_i}}{Z_{\kappa_i,\xi}} - \tau_{\kappa_i} + 2 \leq \overline{T_{\theta_i,\xi}} \qquad \forall i \in N, m \in M \tag{4.162}$$

$$\Rightarrow \frac{\tau_{\theta_i}Z_{\kappa_i,\xi} + \tau_{\kappa_i}Z_{\theta_i,\kappa_i}}{Z_{\theta_i,\kappa_i} \times Z_{\kappa_i,\xi}} - \tau_{\theta_i} - \tau_{\kappa_i} + 2 \leq \overline{T_{\theta_i,\xi}} \qquad \forall i \in N, m \in M \tag{4.163}$$

Let

$$Z_{\theta_i,\kappa_i} \times Z_{\kappa_i,\xi} = D_{\theta_i,\xi} \qquad \forall i \in N \tag{4.164}$$

Apply logarithmic operation on both sides,

$$\Rightarrow \log Z_{\theta_i,\kappa_i} + \log Z_{\kappa_i,\xi} = \log D_{\theta_i,\xi} \qquad \forall i \in N \tag{4.165}$$

$$\epsilon^{n+v+4} \leq D_{\theta_i,\xi} \leq 1 \qquad \forall i \in N \tag{4.166}$$

doi:10.6342/NTU202210101

Replace $Z_{\theta_i,\kappa_i} \times Z_{\kappa_i,\xi}$ with $D_{\theta_i,\xi}$ in formula (4.163).

$$\frac{\tau_{\theta_i} Z_{\kappa_i,\xi} + \tau_{\kappa_i} Z_{\theta_i,\kappa_i}}{D_{\theta_i,\xi}} - \tau_{\theta_i} - \tau_{\kappa_i} + 2 \leq \overline{T_{\theta_i,\xi}} \qquad \forall i \in N \tag{4.167}$$

$$\Rightarrow \tau_{\theta_i} Z_{\kappa_i,\xi} + \tau_{\kappa_i} Z_{\theta_i,\kappa_i} - \tau_{\theta_i} D_{\theta_i,\xi} - \tau_{\kappa_i} D_{\theta_i,\xi} + 2D_{\theta_i,\xi} \leq D_{\theta_i,\xi}\overline{T_{\theta_i,\xi}} \quad \forall i \in N \tag{4.168}$$

As for the throughput constraint (3.34), we replace $q_{\theta_i}(1-q_{\theta_i})^{(v-1)} q_{\kappa R_i} P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ with Z_{θ_i,κ_i} and $q_{\kappa S_i}(1-q_{\kappa S_i})^{(n-1)} q_\xi P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ withZ_{θ_i,κ_i} and $Z_{\kappa_i,\xi}$. We get a new constraint as follow:

$$vZ_{\theta_i,\kappa_i} - Z_{\kappa_i,\xi} \leq 0 \qquad \forall i \in N \tag{4.169}$$

After dealing with the decision variables that multiplies, we relax formula (4.130), formula (4.131), formula (4.157), formula (4.158), formula (4.165), formula (4.168) and formula (4.169) with respective lagrangian multipliers.

The final LR function is derived as the formula below:

doi:10.6342/NTU202210101

$$
\begin{aligned}
\min \quad & \sum_{i\in N}\left(\frac{x_{\theta_i}\times v + y_{\theta_i}\times v + \beta_{\theta_i,\kappa_i}\times v + x_{\kappa_i} + y_{\kappa_i} + \beta_{\kappa_i,\xi} + x_\xi + C^b_\xi(r_\xi,m) - y_\xi}{m}\right)\\
&+\sum_{i\in N}\mu_i^1(\log q_{\theta_i} + (v-1)\log I_{\theta_i} + \log q_{\kappa R_i} + \log P_{\theta_i,\kappa_i}(r_{\theta_i},m) - \log Z_{\theta_i,\kappa_i})\\
&+\sum_{i\in N}\mu_i^2(\log q_{\kappa S_i} + (n-1)\log I_{\kappa S_i} + \log q_\xi + \log P_{\kappa_i,\xi}(r_{\kappa_i},m) - \log Z_{\kappa_i,\xi})\\
&+\sum_{i\in N}\mu_i^3(\log C^a_{\theta_i}(r_{\theta_i},m) + \log q_{\theta_i} - \log x_{\theta_i})\\
&+\sum_{i\in N}\mu_i^4(\log C^a_{\kappa_i}(r_{\kappa_i},m) + \log q_{\kappa R_i} - \log x_{\kappa_i})\\
&+\sum_{i\in N}\mu^5(\log C^a_\xi(r_\xi,m) + \log q_\xi - \log x_\xi)\\
&+\sum_{i\in N}\mu_i^6(\log C^b_{\theta_i}(r_{\theta_i},m) + \log I_{\theta_i} - \log y_{\theta_i})\\
&+\sum_{i\in N}\mu_i^7(\log C^b_{\kappa_i}(r_{\kappa_i},m) + \log I_{\kappa R_i} - \log y_{\kappa_i})\\
&+\sum_{i\in N}\mu^8(\log C^b_\xi(r_\xi,m) + \log q_\xi - \log y_\xi)\\
&+\sum_{i\in N}\mu_i^9(I_{\theta_i} + q_{\theta_i} - 1)\\
&+\sum_{i\in N}\mu_i^{10}(I_{\kappa R_i} + q_{\kappa R_i} - 1)\\
&+\sum_{i\in N}\mu_i^{11}(I_{\kappa S_i} + q_{\kappa S_i} - 1)\\
&+\sum_{i\in N}\mu_i^{12}(\log C^\tau_{\theta_i}(r_{\theta_i},m) + \log q_{\theta_i} - \log \beta_{\theta_i,\kappa_i})\\
&+\sum_{i\in N}\mu_i^{13}(\log C^\tau_{\kappa_i}(r_{\kappa_i},m) + \log q_{\kappa S_i} - \log \beta_{\kappa_i,\xi})\\
&+\sum_{i\in N}\mu_i^{14}(\log Z_{\theta_i,\kappa_i} + \log Z_{\kappa_i,\xi} - \log D_{\theta_i,\xi})\\
&+\sum_{i\in N}\mu_i^{15}(\tau_{\theta_i}Z_{\kappa_i,\xi} + \tau_{\kappa_i}Z_{\theta_i,\kappa_i} - \tau_{\theta_i}D_{\theta_i,\xi} - \tau_{\kappa_i}D_{\theta_i,\xi} + 2D_{\theta_i,\xi} - D_{\theta_i,\xi}\overline{T_{\theta_i,\xi}})\\
&+\sum_{i\in N}\mu_i^{16}(vZ_{\theta_i,\kappa_i} - Z_{\kappa_i,\xi})
\end{aligned}
\tag{4.170}
$$

doi:10.6342/NTU202210101

The objective function is subject to :

$$\epsilon \leq q_{\theta_i} \leq 1 \qquad \forall i \in N \qquad (4.171)$$

$$\epsilon \leq q_{\kappa R_i}, q_{\kappa S_i} \leq 1 \qquad \forall i \in N \qquad (4.172)$$

$$\epsilon \leq q_{\xi} \leq 1 \qquad (4.173)$$

$$0 \leq I_{\theta_i} \leq 1 - \epsilon \qquad \forall i \in N \qquad (4.174)$$

$$0 \leq I_{\kappa R_i} \leq 1 - \epsilon \qquad \forall i \in N \qquad (4.175)$$

$$0 \leq I_{\kappa S_i} \leq 1 - \epsilon \qquad \forall i \in N \qquad (4.176)$$

$$\epsilon \leq P_{\theta_i,\kappa_i}(r_{\theta_i}, m) \leq 1 \qquad \forall i \in N, m \in M \qquad (4.177)$$

$$\epsilon \leq P_{\kappa_i,\xi}(r_{\kappa_i}, m) \leq 1 \qquad \forall i \in N, m \in M \qquad (4.178)$$

$$\epsilon^{v+2} \leq Z_{\theta_i,\kappa_i} \leq 1 \qquad \forall i \in N \qquad (4.179)$$

$$\epsilon^{n+2} \leq Z_{\kappa_i,\xi} \leq 1 \qquad \forall i \in N \qquad (4.180)$$

$$\epsilon^2 \leq x_{\theta_i} \leq C^a_{\theta_i}(r_{\theta_i}, m) \qquad \forall i \in N, m \in M \qquad (4.181)$$

$$\epsilon^2 \leq x_{\kappa_i} \leq C^a_{\kappa_i}(r_{\kappa_i}, m) \qquad \forall i \in N, m \in M \qquad (4.182)$$

$$\epsilon^2 \leq x_{\xi} \leq C^a_{\xi}(r_{\xi}, m) \qquad \forall m \in M \qquad (4.183)$$

$$\epsilon^2 \leq y_{\theta_i} \leq C^b_{\theta_i}(r_{\theta_i}, m) \qquad \forall i \in N, m \in M \qquad (4.184)$$

$$\epsilon^2 \leq y_{\kappa_i} \leq C^b_{\kappa_i}(r_{\kappa_i}, m) \qquad \forall i \in N, m \in M \qquad (4.185)$$

$$\epsilon^2 \leq y_{\xi} \leq C^b_{\xi}(r_{\xi}, m) \qquad \forall m \in M \qquad (4.186)$$

doi:10.6342/NTU202210101

$$C^{\tau}_{\theta_i}(\min R_{\theta_i}, m) \times \epsilon \leq \beta_{\theta_i,\kappa_i} \leq C^{\tau}_{\theta_i}(\max R_{\theta_i}, m) \qquad \forall i \in N, m \in M \tag{4.187}$$

$$C^{\tau}_{\kappa_i}(\min R_{\kappa_i}, m) \times \epsilon \leq \beta_{\kappa_i,\xi} \leq C^{\tau}_{\kappa_i}(\max R_{\kappa_i}, m) \qquad \forall i \in N, m \in M \tag{4.188}$$

$$\epsilon^{n+v+4} \leq D_{\theta_i,\xi} \leq 1 \qquad \forall i \in N \tag{4.189}$$

$$r_{\theta_i} \in R_{\theta_i} \qquad \forall i \in N \tag{4.190}$$

$$r_{\kappa_i} \in R_{\kappa_i} \qquad \forall i \in N \tag{4.191}$$

$$r_{\xi} \in R_{\xi} \tag{4.192}$$

Multipliers $\mu^1, \mu^2, \mu^3, \mu^4, \mu^5, \mu^6, \mu^7, \mu^8, \mu^9, \mu^{10}, \mu^{11}, \mu^{12}, \mu^{13}, \mu^{14}$,μ^{15} and μ^{16} respectively represents the vectors of $\{\mu^1_i\}$, $\{\mu^2_i\}$, $\{\mu^3\}$, $\{\mu^4_i\}$, $\{\mu^5\}$, $\{\mu^6_i\}$, $\{\mu^7_i\}$, $\{\mu^8\}$, $\{\mu^9_i\}$, $\{\mu^{10}_i\}$, $\{\mu^{11}_i\}$, $\{\mu^{12}_i\}$, $\{\mu^{13}_i\}$ $\{\mu^{14}_i\}$, $\{\mu^{15}_i\}$,and $\{\mu^{16}_i\}$. They are either positive or negative due to relaxing equality constraints except μ^{15} and μ^{16}. μ^{15} and μ^{16} are greater or equal to 0 due to relaxing inequality constraints.

4.4.2 The LR Subproblems

4.4.2.1 Subproblem 1(related to decision variable x_{θ_i})

By extracting all decision variables x_{θ_i} in the LR problem, we will then obtain subproblem 1. In subproblem 1, μ^3_i can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 1, we have to take the sign of μ^3_i into consideration. Linear term will not influence the concavity or convexity

doi:10.6342/NTU202210101

of the formula, so μ_i^3 will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ_i^3 is positive with a negative sign in front of it, the formula will become a convex function. And when μ_i^3 is negative, the formula will become a concave function.

The concavity or convexity of subproblem 1 will lead to different solution approaches. When μ_i^3 is positive and therefore changing the coefficient of log term into negative, find the point of x_{θ_i} where the slope is 0, which will be the minimum point of the convex function. If the point of x_{θ_i} where the slope is 0 falls in the legal range, which is the upper and lower bound of x_{θ_i}, then it is the solution to x_{θ_i}. However, if this point does not fall in the legal range of x_{θ_i}, the upper and lower bound of x_{θ_i} will be substitute into the objective function to see whichever is smaller, and it will be the solution to x_{θ_i}.

On the contrary, when μ_i^3 is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to x_{θ_i}.

The minimum of x_{θ_i} occurs when both $C^a_{\theta_i}(r_{\theta_i}, m)$ and q_{θ_i} are at its minimum, which is ϵ. The maximum on the other hand occurs when q_{θ_i} is 1. We separate x_{θ_i} from the objective function and derive a subproblem shown as formula (4.193) and constraint shown as formula (4.194).

doi:10.6342/NTU202210101

Table 4.22: Subproblem 1(related to decision variable x_{θ_i})

Objective function :

$$Z_{sub1} = \min \sum_{i \in N} \left(\frac{x_{\theta_i} \times v}{m} - \mu_i^3 \log x_{\theta_i}\right) \quad (4.193)$$

Subject to :

$$\epsilon^2 \leq x_{\theta_i} \leq C^a_{\theta_i}(r_{\theta_i}, m) \qquad \forall i \in N \quad (4.194)$$

Algorithm 22: Algorithm for Subproblem 1

for *each node i* **do**
  Calculate the objective value at $x_{\theta_i} = \epsilon^2$ denoted as V_1
  Calculate the objective value at $x_{\theta_i} = C^a_{\theta_i}(r_{\theta_i}, m)$ denoted as V_2
  if $\mu_i^3 \geq 0$ **then**
    partial differential to x_{θ_i}
    $\frac{v}{m} - \mu_i^3 \frac{1}{x_{\theta_i} \ln e} = 0$
    $x_{\theta_i} = \frac{\mu_i^3 m}{v}$
    if x_{θ_i} *is not feasible* **then**
      set xtheta[i] to min(V_1, V_2)*, which is the boundary value with smaller objective value*
    end
  else
    set xtheta[i] to min(V_1, V_2)*, which is the boundary value with smaller objective value*
  end
end

4.4.2.2 Subproblem 2(related to decision variable x_{κ_i})

By extracting all decision variables x_{κ_i} in the LR problem, we will then obtain subproblem 2. In subproblem 2, μ_i^4 can be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity of subproblem 2 and the solution

doi:10.6342/NTU202210101

is similar to subproblem 1, we have to take the sign of μ_i^4 into consideration. We separate x_{κ_i} from the objective function and derive a subproblem shown as formula (4.195) and constraint shown as formula (4.196).

Table 4.23: Subproblem 2(related to decision variable x_{κ_i})

Objective function :

$$Z_{sub2} = \min \sum_{i \in N} (\frac{x_{\kappa_i}}{m} - \mu_i^4 \log x_{\kappa_i}) \tag{4.195}$$

Subject to :

$$\epsilon^2 \leq x_{\kappa_i} \leq C^a_{\kappa_i}(r_{\kappa_i}, m) \qquad \forall i \in N \tag{4.196}$$

Algorithm 23: Algorithm for Subproblem 2

for *each node i* **do**
- Calculate the objective value at $x_{\kappa_i} = \epsilon^2$ denoted as V_1
- Calculate the objective value at $x_{\kappa_i} = C^a_{\kappa_i}(r_{\kappa_i}, m)$ denoted as V_2
- **if** $\mu_i^4 \geq 0$ **then**
  - *partial differential to* x_{κ_i}
  - $\frac{1}{m} - \mu_i^4 \frac{1}{x_{\kappa_i} \ln e} = 0$
  - $x_{\kappa_i} = \mu_i^4 m$
  - **if** x_{κ_i} *is not feasible* **then**
    - *set xkappa[i] to* $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  - **end**
- **else**
  - *set xkappa[i] to* $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
- **end**

end

doi:10.6342/NTU202210101

4.4.2.3 Subproblem 3(related to decision variable x_ξ)

By extracting all decision variables x_ξ in the LR problem, we will then obtain subproblem 3. In subproblem 3, μ_i^5 can be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity of subproblem 3 and the solution is similar to subproblem 1, we have to take the sign of μ_i^5 into consideration. We separate x_ξ from the objective function and derive a subproblem shown as formula (4.197) and constraint shown as formula (4.198).

Table 4.24: Subproblem 3(related to decision variable x_ξ)

Objective function :

$$Z_{sub3} = \min \sum_{i \in N} (\frac{x_\xi}{m} - \mu^5 \log x_\xi) \tag{4.197}$$

Subject to :

$$\epsilon^2 \leq x_\xi \leq C_\xi^a(r_\xi, m) \tag{4.198}$$

doi:10.6342/NTU202210101

Algorithm 24: Algorithm for Subproblem 3

for *each node i* **do**
  Calculate the objective value at $x_\xi = \epsilon^2$ denoted as V_1
  Calculate the objective value at $x_\xi = C^a_\xi(r_\xi, m)$ denoted as V_2
  if $\mu^5_i \geq 0$ **then**
    partial differential to x_ξ
    $\frac{1}{m} - \mu^5 \frac{1}{x_\xi \ln e} = 0$
    $x_\xi = \mu^5 m$
    if x_ξ *is not feasible* **then**
      set xxi[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set xxi[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.4.2.4 Subproblem 4(related to decision variable y_{θ_i})

By extracting all decision variables y_{θ_i} in the LR problem, we will then obtain subproblem 4. In subproblem 4, μ^6_i can be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity of subproblem 6 and the solution is similar to subproblem 1, we have to take the sign of μ^6_i into consideration. We separate y_{θ_i} from the objective function and derive a subproblem shown as formula (4.199) and constraint shown as formula (4.200).

doi:10.6342/NTU202210101

Table 4.25: Subproblem 4(related to decision variable y_{θ_i})

Objective function :

$$Z_{sub4} = \min \sum_{i \in N} (\frac{y_{\theta_i} \times v}{m} - \mu_i^6 \log y_{\theta_i}) \quad (4.199)$$

Subject to :

$$\epsilon^2 \leq y_{\theta_i} \leq C_{\theta_i}^b(r_{\theta_i}, m) \qquad \forall i \in N \quad (4.200)$$

Algorithm 25: Algorithm for Subproblem 4

for *each node i* **do**
    Calculate the objective value at $y_{\theta_i} = \epsilon^2$ denoted as V_1
    Calculate the objective value at $y_{\theta_i} = C_{\theta_i}^b(r_{\theta_i}, m)$ denoted as V_2
    if $\mu_i^6 \geq 0$ **then**
        partial differential to y_{θ_i}
        $\frac{v}{m} - \mu_i^6 \frac{1}{y_{\theta_i} \ln e} = 0$
        $y_{\theta_i} = \frac{\mu_i^6 m}{v}$
        if y_{θ_i} *is not feasible* **then**
            set ytheta[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
        end
    else
        set ytheta[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
end

4.4.2.5 Subproblem 5(related to decision variable y_{κ_i})

By extracting all decision variables y_{κ_i} in the LR problem, we will then obtain subproblem 5. In subproblem 5, μ_i^7 can be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity of subproblem 5 and the solution

doi:10.6342/NTU202210101

is similar to subproblem 1, we have to take the sign of μ_i^7 into consideration. We separate y_{κ_i} from the objective function and derive a subproblem shown as formula (4.201) and constraint shown as formula (4.202).

Table 4.26: Subproblem 5(related to decision variable y_{κ_i})

Objective function :

$$Z_{sub5} = \min \sum_{i \in N} (\frac{y_{\kappa_i}}{m} - \mu_i^7 \log y_{\kappa_i}) \quad (4.201)$$

Subject to :

$$\epsilon^2 \le y_{\kappa_i} \le C^b_{\kappa_i}(r_{\kappa_i}, m) \qquad \forall i \in N \quad (4.202)$$

Algorithm 26: Algorithm for Subproblem 5

for *each node i* **do**
  Calculate the objective value at $y_{\kappa_i} = \epsilon^2$ denoted as V_1
  Calculate the objective value at $y_{\kappa_i} = C^b_{\kappa_i}(r_{\kappa_i}, m)$ denoted as V_2
  if $\mu_i^7 \ge 0$ **then**
    partial differential to y_{κ_i}
    $\frac{1}{m} - \mu_i^7 \frac{1}{y_{\kappa_i} \ln e} = 0$
    $y_{\kappa_i} = \mu_i^7 m$
    if y_{κ_i} *is not feasible* **then**
      set ykappa[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set ykappa[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

doi:10.6342/NTU202210101

4.4.2.6 Subproblem 6(related to decision variable y_ξ)

By extracting all decision variables y_ξ in the LR problem, we will then obtain subproblem 6. In subproblem 6, μ_i^8 can be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity of subproblem 6 and the solution is similar to subproblem 1, we have to take the sign of μ_i^8 into consideration. We separate y_ξ from the objective function and derive a subproblem shown as formula (4.203) and constraint shown as formula (4.204).

Table 4.27: Subproblem 6(related to decision variable y_ξ)

Objective function :

$$Z_{sub6} = \min \sum_{i \in N} (-\frac{y_\xi}{m} - \mu_i^8 \log y_\xi) \tag{4.203}$$

Subject to :

$$\epsilon^2 \leq y_\xi \leq C_\xi^b(r_\xi, m) \tag{4.204}$$

doi:10.6342/NTU202210101

Algorithm 27: Algorithm for Subproblem 6

for *each node i* **do**
  Calculate the objective value at $y_\xi = \epsilon^2$ denoted as V_1
  Calculate the objective value at $y_\xi = C_\xi^b(r_\xi, m)$ denoted as V_2
  if $\mu_i^8 \geq 0$ **then**
    partial differential to y_ξ
    $-\frac{1}{m} - \mu_i^8 \frac{1}{y_\xi \ln e} = 0$
    $y_\xi = -\mu_i^8 m$
    if y_ξ *is not feasible* **then**
      set yxi[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set yxi[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.4.2.7 Subproblem 7(related to decision variable q_{θ_i})

By extracting all decision variables q_{θ_i} in the LR problem, we will then obtain subproblem 7. In subproblem 7, there are 4 multipliers μ_i^1,μ_i^3,$\mu^9$and μ_i^{12} and all of them can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 7, we have to take the sign of μ_i^1,μ_i^3 and μ_i^{12}into consideration, but not μ^9 because q_{θ_i} is linear so it won't affect the concavity or convexity of subproblem 7. A log function by itself is a concave function, so μ_i^1,μ_i^3 and μ_i^{12} can all play an important role in deciding whether the function is concave or convex.

In this section, we developed a way to easily get the convexity or concavity of this function. We first find the extreme point by finding points with derivative = 0 and will

doi:10.6342/NTU202210101

obtain a objective value for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of q_{θ_i}. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of q_{θ_i}, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of q_{θ_i}, if yes then we can return it as our solution to q_{θ_i}. If the point with derivative = 0 doesn't appears in the legal range of q_{θ_i} and is considered infeasible, we will return the boundary point of q_{θ_i} with a smaller objective value as our solution to q_{θ_i}.

On the contrary, if the objective value with the point of derivative = 0 is bigger than the objective value of the boundary points of q_{θ_i}, then the function is concave. Therefore, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution to q_{θ_i}.

We separate q_{θ_i} from the objective function and derive a subproblem shown as formula (4.205) and constraint shown as formula (4.206).

Table 4.28: Subproblem 7(related to decision variable q_{θ_i})

Objective function :

$$Z_{sub7} = \min \sum_{i \in N} (\mu_i^1 \log q_{\theta_i} + \mu_i^3 \log q_{\theta_i} + \mu_i^9 q_{\theta_i} + \mu_i^{12} \log q_{\theta_i}) \tag{4.205}$$

Subject to :

$$\epsilon \le q_{\theta_i} \le 1 \qquad \forall i \in N \tag{4.206}$$

doi:10.6342/NTU202210101

Algorithm 28: Algorithm for Subproblem 7

for *each node i* **do**
  Calculate the objective value at $q_{\theta_i} = \epsilon$ denoted as V_1
  Calculate the objective value at $q_{\theta_i} = 1$ denoted as V_2
  partial differential to q_{θ_i}
  $\mu_i^1 \frac{1}{q_{\theta_i} \ln e} + \mu_i^3 \frac{1}{q_{\theta_i} \ln e} + \mu_i^9 + \mu_i^{12} \frac{1}{q_{\theta_i} \ln e} = 0$
  $\frac{\mu_i^1 + \mu_i^3 + \mu_i^{12}}{q_{\theta_i}} = -\mu_i^9$
  $q_{\theta_i} = -\frac{\mu_i^1 + \mu_i^3 + \mu_i^{12}}{\mu_i^9}$
  if $Z_{sub7}(q_{\theta_i}) \leq \min(V_1, V_2)$ **then**
    $Z_{sub7} = convex$
    if q_{θ_i} *is feasible* **then**
      qtheta[i] = q_{θ_i}
    else
      set qtheta[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
  else
    set qtheta[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  end
end

4.4.2.8 Subproblem 8(related to decision variable $q_{\kappa R_i}$)

By extracting all decision variables $q_{\kappa R_i}$ in the LR problem, we will then obtain subproblem 8. In subproblem 8, there are 3 multipliers μ_i^1,μ_i^4 and μ^{10} and all of them can be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity and the solution to subproblem 8 is similar to subproblem 7. We have to take the sign of μ_i^1,μ_i^4 and μ^{10}into consideration, but not μ^{10} because $q_{\kappa R_i}$ is linear so it won't affect the concavity or convexity of subproblem 8.

We separate $q_{\kappa R_i}$ from the objective function and derive a subproblem shown as for-

doi:10.6342/NTU202210101

mula (4.207) and constraint shown as formula (4.208).

Table 4.29: Subproblem 8(related to decision variable $q_{\kappa R_i}$)

Objective function : $$Z_{sub8} = \min \sum_{i \in N} (\mu_i^1 \log q_{\kappa R_i} + \mu_i^4 \log q_{\kappa R_i} + \mu_i^{10} q_{\kappa R_i}) \quad (4.207)$$ **Subject to :** $$\epsilon \leq q_{\kappa R_i} \leq 1 \qquad \forall i \in N \quad (4.208)$$

Algorithm 29: Algorithm for Subproblem 8

for *each node i* **do**
- Calculate the objective value at $q_{\kappa R_i} = \epsilon$ denoted as V_1
- Calculate the objective value at $q_{\kappa R_i} = 1$ denoted as V_2
- partial differential to $q_{\kappa R_i}$
- $\mu_i^1 \frac{1}{q_{\kappa R_i} \ln e} + \mu_i^4 \frac{1}{q_{\kappa R_i} \ln e} + \mu_i^{10} = 0$
- $\frac{\mu_i^1 + \mu_i^4}{q_{\kappa R_i}} = -\mu_i^{10}$
- $q_{\kappa R_i} = -\frac{\mu_i^1 + \mu_i^4}{\mu_i^{10}}$
- **if** $Z_{sub8}(q_{\kappa R_i}) \leq \min(V_1, V_2)$ **then**
  - Z$_{sub8} = convex$
  - **if** $q_{\kappa R_i}$ *is feasible* **then**
    - qkappaR[i] = $q_{\kappa R_i}$
  - **else**
    - set qkappaR[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  - **end**
- **else**
  - set qkappaR[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
- **end**

end

doi:10.6342/NTU202210101

4.4.2.9 Subproblem 9(related to decision variable q_ξ)

By extracting all decision variables q_ξ in the LR problem, we will then obtain subproblem 9. In subproblem 9, there are 3 multipliers μ_i^2,$\mu^5$and μ_i^8 and all of them can be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity and the solution to subproblem 9 is similar to subproblem 7. We have to take the sign of μ_i^2,$\mu^5$and μ_i^8 into consideration.

We separate q_ξ from the objective function and derive a subproblem shown as formula (4.209) and constraint shown as formula (4.210).

Table 4.30: Subproblem 9(related to decision variable q_ξ)

Objective function :

$$Z_{sub9} = \min \sum_{i \in N} (\mu_i^2 \log q_\xi + \mu^5 \log q_\xi + \mu^8 \log q_\xi) \tag{4.209}$$

Subject to :

$$\epsilon \leq q_\xi \leq 1 \tag{4.210}$$

doi:10.6342/NTU202210101

Algorithm 30: Algorithm for Subproblem 9

for *each node i* **do**
  Calculate the objective value at $q_\xi = \epsilon$ denoted as V_1
  Calculate the objective value at $q_\xi = 1$ denoted as V_2
  partial differential to q_ξ
  $\mu_i^2 \frac{1}{q_\xi \ln e} + \mu^5 \frac{1}{q_\xi \ln e} + \mu^8 \frac{1}{q_\xi \ln e} = 0$
  $\frac{\mu_i^2 + \mu^5 + \mu^8}{q_\xi} = 0$
  get solution to q_ξ
  if $Z_{sub9}(q_\xi) \leq \min(V_1, V_2)$ **then**
    $Z_{sub9} = convex$
    if q_ξ *is feasible* **then**
      qxi[i] = q_ξ
    else
      set qxi[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
  else
    set qxi[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  end
end

4.4.2.10 Subproblem 10(related to decision variable r_{θ_i})

By extracting all decision variables r_{θ_i} in the LR problem, we will then obtain subproblem 10. In subproblem 10, there are 4 multipliers μ_i^1, μ_i^3, μ_i^6 and μ_i^{12}. All of them can be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity and the solution to subproblem 10 is similar to subproblem 7. We have to take the sign of μ_i^1, μ_i^3, μ_i^6 and μ_i^{12} into consideration.

We separate r_{θ_i} from the objective function and derive a subproblem shown as formula (4.211) and constraint shown as formula (4.212) and formula (4.213).

doi:10.6342/NTU202210101

Table 4.31: Subproblem 10(related to decision variable r_θ)

Objective function :

$$Z_{sub10} = \min \sum_{i \in N} \Big[\mu_i^1 \log P_{\theta_i,\kappa_i}(r_{\theta i}, m) + \mu_i^3 \log C^a_{\theta_i}(r_{\theta_i}, m) + \mu_i^6 \log C^b_{\theta_i}(r_{\theta_i}, m) + \mu_i^{12} \log C^\tau_{\theta_i}(r_{\theta_i}, m) \Big] \tag{4.211}$$

Subject to :

$$\epsilon \le P_{\theta_i,\kappa_i}(r_{\theta i}, m) \le 1 \qquad \forall i \in N \tag{4.212}$$

$$r_{\theta_i} \in R_{\theta_i} \qquad \forall i \in N \tag{4.213}$$

Algorithm 31: Algorithm for Subproblem 10

for *each node i* **do**
  Calculate the objective value at $r_{\theta_i} = \max R_{\theta_i}$ denoted as V_1
  Calculate the objective value at $r_{\theta_i} = \min R_{\theta_i}$ denoted as V_2
  Get solution of r_{θ_i} where Z_{sub10} has the smallest value
  if $Z_{sub10}(r_{\theta_i}, m) \le \min(V_1, V_2)$ **then**
    $Z_{sub10} = convex$
    if r_{θ_i} *is feasible* **then**
      rtheta[i] = r_{θ_i}
    else
      set rtheta[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set rtheta[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.4.2.11 Subproblem 11(related to decision variable r_{κ_i})

By extracting all decision variables r_{κ_i} in the LR problem, we will then obtain subproblem 11. In subproblem 11, there are 4 multipliers μ_i^2, μ_i^4, μ_i^7 and μ_i^{13}. All of them can

doi:10.6342/NTU202210101

be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity and the solution to subproblem 11 is similar to subproblem 7. We have to take the sign of μ_i^2, μ_i^4, μ_i^7 and μ_i^{13} into consideration.

We separate r_{κ_i} from the objective function and derive a subproblem shown as formula (4.214) and constraint shown as formula (4.215) and formula (4.216).

Table 4.32: Subproblem 11(related to decision variable r_κ)

Objective function :

$$Z_{sub11} = \min \sum_{i \in N} \Big[\mu_i^2 \log P_{\kappa_i,\xi}(r_{\kappa_i}, m) + \mu_i^4 \log C^a_{\kappa_i}(r_{\kappa_i}, m) + \mu_i^7 \log C^b_{\kappa_i}(r_{\kappa_i}, m) + \mu_i^{13} \log C^\tau_{\kappa_i}(r_{\kappa_i}, m)\Big] \tag{4.214}$$

Subject to :

$$\epsilon \le P_{\kappa_i,\xi}(r_{\kappa_i}, m) \le 1 \qquad \forall i \in N \tag{4.215}$$

$$r_{\kappa_i} \in R_{\kappa_i} \qquad \forall i \in N \tag{4.216}$$

doi:10.6342/NTU202210101

Algorithm 32: Algorithm for Subproblem 11

for *each node i* **do**
  Calculate the objective value at $r_{\kappa_i} = \max R_{\kappa_i}$ denoted as V_1
  Calculate the objective value at $r_{\kappa_i} = \min R_{\kappa_i}$ denoted as V_2
  Get solution of r_{κ_i} where Z_{sub11} has the smallest value
  if $Z_{sub11}(r_{\kappa_i}, m) \leq \min(V_1, V_2)$ **then**
    $Z_{sub11} = convex$
    if r_{κ_i} *is feasible* **then**
      rkappa[i] = r_{κ_i}
    else
      set rkappa[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set rkappa[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.4.2.12 Subproblem 12(related to decision variable r_ξ)

By extracting all decision variables r_ξ in the LR problem, we will then obtain subproblem 12. In subproblem 12, there are 2 multipliers μ^5 and μ^8. All of them can be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity and the solution to subproblem 12 is similar to subproblem 7. We have to take the sign of μ^5 and μ^8 into consideration.

We separate r_ξ from the objective function and derive a subproblem shown as formula (4.217) and constraint shown as formula (4.218).

doi:10.6342/NTU202210101

Table 4.33: Subproblem 12(related to decision variable r_ξ)

Objective function :

$$Z_{sub12} = \min \sum_{i \in N} \Big[C_\xi^b(r_\xi, m) + \mu^5 \log C_\xi^a(r_\xi, m) + \mu^8 \log C_\xi^b(r_\xi, m) \Big] \tag{4.217}$$

Subject to :

$$r_\xi \in R_\xi \tag{4.218}$$

Algorithm 33: Algorithm for Subproblem 12

```
for each node i do
    Calculate the objective value at r_ξ = max R_ξ denoted as V_1
    Calculate the objective value at r_ξ = min R_ξ denoted as V_2
    Get solution of r_ξ where Z_sub12 has the smallest value
    if Z_sub12(r_ξ) ≤ min(V_1, V_2) then
        Z_sub12 = convex
        if r_ξ is feasible then
            rxi = r_ξ
        else
            set rxi to min(V_1, V_2), which is the boundary value with smaller
              objective value
        end
    else
        set rxi to min(V_1, V_2), which is the boundary value with smaller
          objective value
    end
end
```

4.4.2.13 Subproblem 13(related to decision variable Z_{θ_i,κ_i})

By extracting all decision variables Z_{θ_i,κ_i} in the LR problem, we will then obtain subproblem 13. In subproblem 13, there are 4 multipliers μ_i^1,μ_i^{14},μ_i^{14} and μ_i^{16}. μ_i^1 and μ_i^{14} can be either positive or negative due to relaxing equality constraints and μ_i^{15} and μ_i^{16} should be greater or equal to positive due to relaxing inequality constraints.

doi:10.6342/NTU202210101

The way to determine the concavity or convexity and the solution to subproblem 13 is similar to subproblem 7. We have to take the sign of μ_i^1,μ_i^{14},μ_i^{15} and μ_i^{16} into consideration.

We separate Z_{θ_i,κ_i} from the objective function and derive a subproblem shown as formula (4.219) and constraint shown as formula (4.220).

Table 4.34: Subproblem 13(related to decision variable Z_{θ_i,κ_i})

Objective function :

$$Z_{sub13} = \min \sum_{i \in N} \Big[-\mu_i^1 \log Z_{\theta_i,\kappa_i} + \mu_i^{14} \log Z_{\theta_i,\kappa_i} + \mu_i^{15} \tau_{\kappa_i} Z_{\theta_i,\kappa_i} + \mu_i^{16} \upsilon Z_{\theta_i,\kappa_i} \Big] \tag{4.219}$$

Subject to :

$$\epsilon^{\upsilon+2} \leq Z_{\theta_i,\kappa_i} \leq 1 \qquad \forall i \in N \tag{4.220}$$

doi:10.6342/NTU202210101

Algorithm 34: Algorithm for Subproblem 13

for *each node i* **do**
  Calculate the objective value at $Z_{\theta_i,\kappa_i} = \epsilon^{v+2}$ denoted as V_1
  Calculate the objective value at $Z_{\theta_i,\kappa_i} = 1$ denoted as V_2
  partial differential to Z_{θ_i,κ_i}
  $-\mu_i^1 \frac{1}{Z_{\theta_i,\kappa_i} \ln e} + \mu_i^{14} \frac{1}{Z_{\theta_i,\kappa_i} \ln e} + \mu_i^{15}\tau_{\kappa_i} + \mu_i^{16} v = 0$
  $-\mu_i^1 + \mu_i^{14} + Z_{\theta_i,\kappa_i}\mu_i^{15}\tau_{\kappa_i} + Z_{\theta_i,\kappa_i}\mu_i^{16} v = 0$
  $Z_{\theta_i,\kappa_i} = \frac{\mu_i^1 - \mu_i^{14}}{\mu_i^{15}\tau_{\kappa_i} + \mu_i^{16} v}$
  if $Z_{sub13}(Z_{\theta_i,\kappa_i}) \leq \min(V_1, V_2)$ **then**
    $Z_{sub13} = convex$
    if Z_{θ_i,κ_i} *is feasible* **then**
      ztheta[i] = Z_{θ_i,κ_i}
    else
      set ztheta[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
  else
    set ztheta[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  end
end

4.4.2.14 Subproblem 14(related to decision variable $Z_{\kappa_i,\xi}$)

By extracting all decision variables $Z_{\kappa_i,\xi}$ in the LR problem, we will then obtain subproblem 14. In subproblem 14, there are 4 multipliers μ_i^2,μ_i^{14},μ_i^{15} and μ_i^{16}. μ_i^2 and μ_i^{14} can be either positive or negative due to relaxing equality constraints and μ_i^{15} and μ_i^{16} should be greater or equal to positive due to relaxing inequality constraints.

The way to determine the concavity or convexity and the solution to subproblem 14 is similar to subproblem 7. We have to take the sign of μ_i^2,μ_i^{14},μ_i^{15} and μ_i^{16} into consideration.

We separate $Z_{\kappa_i,\xi}$ from the objective function and derive a subproblem shown as formula (4.221) and constraint shown as formula (4.222).

doi:10.6342/NTU202210101

Table 4.35: Subproblem 14(related to decision variable $Z_{\kappa_i,\xi}$)

Objective function :

$$Z_{sub14} = \min \sum_{i \in N} \Big[-\mu_i^2 \log Z_{\kappa_i,\xi} + \mu_i^{14} \log Z_{\kappa_i,\xi} + \mu_i^{15} \tau_{\theta_i} Z_{\kappa_i,\xi} - \mu_i^{16} Z_{\kappa_i,\xi} \Big] \tag{4.221}$$

Subject to :

$$\epsilon^{n+2} \leq Z_{\kappa_i,\xi} \leq 1 \qquad \forall i \in N \tag{4.222}$$

Algorithm 35: Algorithm for Subproblem 14

```
for each node i do
    Calculate the objective value at Z_{κ_i,ξ} = ε^{n+2} denoted as V_1
    Calculate the objective value at Z_{κ_i,ξ} = 1 denoted as V_2
    partial differential to Z_{κ_i,ξ}
    −μ_i^2 · 1/(Z_{κ_i,ξ} ln e) + μ_i^14 · 1/(Z_{κ_i,ξ} ln e) + μ_i^15 τ_{θ_i} − μ_i^16 = 0
    −μ_i^2 + μ_i^14 + Z_{κ_i,ξ} μ_i^15 τ_{θ_i} − Z_{κ_i,ξ} μ_i^16 = 0
    Z_{κ_i,ξ} = (μ_i^2 − μ_i^14) / (μ_i^15 τ_{θ_i} − μ_i^16)
    if Z_sub14(Z_{κ_i,ξ}) ≤ min(V_1, V_2) then
        Z_sub14 = convex
        if Z_{κ_i,ξ} is feasible then
            zkappa[i] = Z_{κ_i,ξ}
        else
            set zkappa[i] to min(V_1, V_2), which is the boundary value with
              smaller objective value
        end
    else
        set zkappa[i] to min(V_1, V_2), which is the boundary value with smaller
          objective value
    end
end
```

4.4.2.15 Subproblem 15(related to decision variable $\beta_{\theta_i,\kappa_i}$)

The procedure of dividing $\beta_{\theta_i,\kappa_i}$ from the LR problem is similar with subproblem 1.

By extracting all decision variables $\beta_{\theta_i,\kappa_i}$ in the LR problem, we will then obtain subprob-

doi:10.6342/NTU202210101

lem 15. In subproblem 15, μ_i^{12} can be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 15, we have to take the sign of μ_i^{12} into consideration. Linear term will not influence the concavity or convexity of the formula, so μ_i^{12} will singly decide whether it is concave or convex. A log function by itself is a concave function, so when μ_i^{12} is positive with a negative sign in front of it, the formula will become a convex function. And when μ_i^{12} is negative, the formula will become a concave function.

The concavity or convexity of subproblem 15 will lead to different solution approaches. When μ_i^{12} is positive and therefore changing the coefficient of log term into negative, find the point of $\beta_{\theta_i,\kappa_i}$ where the slope is 0, which will be the minimum point of the convex function. If the point of $\beta_{\theta_i,\kappa_i}$ where the slope is 0 falls in the legal range, which is the upper and lower bound of $\beta_{\theta_i,\kappa_i}$, then it is the solution to $\beta_{\theta_i,\kappa_i}$. However, if this point does not fall in the legal range of $\beta_{\theta_i,\kappa_i}$, the upper and lower bound of $\beta_{\theta_i,\kappa_i}$ will be substitute into the objective function to see whichever is smaller, and it will be the solution to $\beta_{\theta_i,\kappa_i}$.

On the contrary, when μ_i^{12} is negative and therefore changing the coefficient of log term into positive, then the function is concave. Hence, we can compare the objective values of the boundary points and the one with smaller objective value will be our solution

doi:10.6342/NTU202210101

to μ_i^{12}.

We separate $\beta_{\theta_i,\kappa_i}$ from the objective function and derive a subproblem shown as formula (4.223) and constraint shown as formula (4.224).

Table 4.36: Subproblem 15(related to decision variable $\beta_{\theta_i,\kappa_i}$)

Objective function :

$$Z_{sub15} = \min \sum_{i \in N} (\frac{\beta_{\theta_i,\kappa_i} v}{m} - \mu_i^{12} \log \beta_{\theta_i,\kappa_i}) \tag{4.223}$$

Subject to :

$$C_{\theta_i}^{\tau}(\min R_i, m) \times \epsilon \leq \beta_{\theta_i,\kappa_i} \leq C_{\theta_i}^{\tau}(\max R_i, m) \qquad \forall i \in N, m \in M \tag{4.224}$$

Algorithm 36: Algorithm for Subproblem 15

for *each node i* **do**
  Calculate the objective value at $\beta_{\theta_i,\kappa_i} = C_{\theta_i}^{\tau}(\min R_i, m) \times \epsilon$ denoted as V_1
  Calculate the objective value at $\beta_{\theta_i,\kappa_i} = C_{\theta_i}^{\tau}(\max R_i, m)$ denoted as V_2
  if $\mu_i^{12} \geq 0$ **then**
    partial differential to $\beta_{\theta_i,\kappa_i}$
    $\frac{v}{m} - \mu_i^{12} \frac{1}{\beta_{\theta_i,\kappa_i} \ln e} = 0$
    $\beta_{\theta_i,\kappa_i} = \frac{\mu_i^{12} m}{v}$
    if $\beta_{\theta_i,\kappa_i}$ *is not feasible* **then**
      set betatheta[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set betatheta[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

doi:10.6342/NTU202210101

4.4.2.16 Subproblem 16(related to decision variable $\beta_{\kappa_i,\xi}$)

The procedure of dividing $\beta_{\kappa_i,\xi}$ from the LR problem and the way to determine the concavity or covexity is similar to subproblem 1. By extracting all decision variables $\beta_{\kappa_i,\xi}$ in the LR problem, we will then obtain subproblem 16. In subproblem 16, μ_i^{13} can be either positive or negative due to relaxing equality constraints.

We separate $\beta_{\kappa_i,\xi}$ from the objective function and derive a subproblem shown as formula (4.225) and constraint shown as formula (4.226).

Table 4.37: Subproblem 16(related to decision variable $\beta_{\kappa_i,\xi}$)

Objective function :

$$Z_{sub16} = \min \sum_{i \in N} (\frac{\beta_{\kappa_i,\xi}}{m} - \mu_i^{13} \log \beta_{\kappa_i,\xi}) \tag{4.225}$$

Subject to :

$$C^{\tau}_{\kappa_i}(\min R_i, m) \times \epsilon \leq \beta_{\kappa_i,\xi} \leq C^{\tau}_{\kappa_i}(\max R_i, m) \qquad \forall i \in N, m \in M \tag{4.226}$$

doi:10.6342/NTU202210101

Algorithm 37: Algorithm for Subproblem 16

for *each node i* **do**
  Calculate the objective value at $\beta_{\kappa_i,\xi} = C^{\tau}_{\kappa_i}(\min R_i, m) \times \epsilon$ denoted as V_1
  Calculate the objective value at $\beta_{\kappa_i,\xi} = C^{\tau}_{\kappa_i}(\max R_i, m)$ denoted as V_2
  if $\mu_i^{13} \geq 0$ **then**
    partial differential to $\beta_{\kappa_i,\xi}$
    $\frac{1}{m} - \mu_i^{13}\frac{1}{\beta_{\kappa_i,\xi}\ln e} = 0$
    $\beta_{\kappa_i,\xi} = \mu_i^{13} m$
    if $\beta_{\kappa_i,\xi}$ *is not feasible* **then**
      set betakappa[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
    end
  else
    set betakappa[i] to $\min(V_1, V_2)$*, which is the boundary value with smaller objective value*
  end
end

4.4.2.17 Subproblem 17(related to decision variable I_{θ_i})

By extracting all decision variables I_{θ_i} in the LR problem, we will then obtain subproblem 17. In subproblem 17, there are 3 multipliers μ_i^1,μ_i^6 and μ_i^9. μ_i^1,μ_i^6 and μ_i^9 can all be either positive or negative due to relaxing equality constraints.

In order to determine the concavity or convexity of subproblem 17, we have to take all multipliers μ_i^1 and μ_i^6 into consideration. Linear term will not influence the concavity or convexity of the formula so we don't have to look at μ_i^9. A log function by itself is a concave function, so μ_i^1 and μ_i^6 will both play an important role in deciding whether it is concave or convex.

To easily and effectively get the convexity or concavity of this function, we first find the extreme point by finding points with derivative = 0 and will obtain a objective value

doi:10.6342/NTU202210101

for it. Later we can compare the objective value with the point of derivative = 0 with the objective value of the boundary points of I_{θ_i}. If the objective value with the point of derivative = 0 is smaller than the objective values of the boundary points of I_{θ_i}, then the function is a convex function. We can then examine whether the point with derivative = 0 falls in the legal range of I_{θ_i}, if yes then we can return it as our solution to I_{θ_i}. If the point with derivative = 0 doesn't appears in the legal range of I_{θ_i} and is considered infeasible, we will return the boundary point of I_{θ_i} with a smaller objective value as our solution to I_{θ_i}.

We separate I_{θ_i} from the objective function and derive a subproblem shown as formula (4.227) and constraint shown as formula (4.228) .

Table 4.38: Subproblem 17(related to decision variable I_{θ_i})

Objective function :

$$Z_{sub17} = \min \sum_{i \in N} (\mu_i^1 (v-1) \log I_{\theta_i} + \mu_i^6 \log I_{\theta_i} + \mu_i^9 I_{\theta_i}) \tag{4.227}$$

Subject to :

$$0 \leq I_{\theta_i} \leq 1 - \epsilon \qquad \forall i \in N \tag{4.228}$$

4.4.2.18 Subproblem 18(related to decision variable $I_{\kappa R_i}$)

By extracting all decision variables $I_{\kappa R_i}$ in the LR problem, we will then obtain subproblem 18. In subproblem 18, there are 2 multipliers μ_i^7 and μ_i^{10}. μ_i^7 and μ_i^{10} can both be

doi:10.6342/NTU202210101

Algorithm 38: Algorithm for Subproblem 17

for *each node i* **do**
    Calculate the objective value at $I_{\theta_i} = 0$ denoted as V_1
    Calculate the objective value at $I_{\theta_i} = 1 - \epsilon$ denoted as V_2
    partial differential to I_{θ_i}
    $\mu_i^1 \frac{v-1}{I_{\theta_i} \ln e} + \mu_i^6 \frac{1}{I_{\theta_i} \ln e} + \mu_i^9 = 0$
    $I_{\theta_i} = -\frac{(v-1)\mu_i^1 + \mu_i^6}{\mu_i^9}$
    if $Z_{sub17}(I_{\theta_i}) \leq \min(V_1, V_2)$ **then**
        $Z_{sub17} = convex$
        if I_{θ_i} *is feasible* **then**
            Itheta[i] = I_{θ_i}
        else
            set Itheta[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
        end
    else
        set Itheta[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
end

either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity of subproblem 18 and the solution is similar to subproblem 17. We separate $I_{\kappa R_i}$ from the objective function and derive a subproblem shown as formula (4.229) and constraint shown as formula (4.230) .

Table 4.39: Subproblem 18(related to decision variable $I_{\kappa R_i}$)

Objective function :

$$Z_{sub18} = \min \sum_{i \in N} (\mu_i^7 \log I_{\kappa R_i} + \mu_i^{10} I_{\kappa R_i}) \tag{4.229}$$

Subject to :

$$0 \leq I_{\kappa R_i} \leq 1 - \epsilon \qquad \forall i \in N \tag{4.230}$$

doi:10.6342/NTU202210101

Algorithm 39: Algorithm for Subproblem 18

for *each node i* **do**
  Calculate the objective value at $I_{\kappa R_i} = 0$ denoted as V_1
  Calculate the objective value at $I_{\kappa R_i} = 1 - \epsilon$ denoted as V_2
  partial differential to $I_{\kappa R_i}$
  $\mu_i^7 \frac{1}{I_{\kappa R_i} \ln e} + \mu_i^{10} = 0$
  $I_{\kappa R_i} = -\frac{\mu_i^7}{\mu_i^{10}}$
  if $Z_{sub18}(I_{\kappa R_i}) \leq \min(V_1, V_2)$ **then**
    $Z_{sub18} = convex$
    if $I_{\kappa R_i}$ *is feasible* **then**
      IkappaR[i] = $I_{\kappa R_i}$
    else
      set IkappaR[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
  else
    set IkappaR[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  end
end

4.4.2.19 Subproblem 19(related to decision variable $D_{\theta_i,\xi}$)

By extracting all decision variables $D_{\theta_i,\xi}$ in the LR problem, we will then obtain subproblem 19. In subproblem 19, there are 2 multipliers μ_i^{15} and μ_i^{16}. μ_i^{15} and μ_i^{16} should be greater or equal to positive due to relaxing inequality constraints.

We separate $D_{\theta_i,\xi}$ from the objective function and derive a subproblem shown as formula (4.231) and constraint shown as formula (4.232).

doi:10.6342/NTU202210101

Table 4.40: Subproblem 19(related to decision variable $D_{\theta_i,\xi}$)

Objective function :

$$Z_{sub19} = \min \sum_{i \in N} \Big[-\mu_i^{14} \log D_{\theta_i,\xi} - \mu_i^{15} \tau_{\theta_i} D_{\theta_i,\xi} - \mu_i^{15} \tau_{\kappa_i} D_{\theta_i,\xi} + 2\mu_i^{15} D_{\theta_i,\xi} - \mu_i^{15} D_{\theta_i,\xi} \overline{T_{\theta_i,\xi}} \Big] \quad (4.231)$$

Subject to :

$$\epsilon^{n+v+2} \leq D_{\theta_i,\xi} \leq 1 \qquad \forall i \in N \quad (4.232)$$

Algorithm 40: Algorithm for Subproblem 19

for *each node i* **do**
- Calculate the objective value at $D_{\theta_i,\xi} = \epsilon^{n+v+2}$ denoted as V_1
- Calculate the objective value at $D_{\theta_i,\xi} = 1$ denoted as V_2
- partial differential to $D_{\theta_i,\xi}$
- $-\mu_i^{14} \frac{1}{D_{\theta_i,\xi} \ln e} - \mu_i^{15} \tau_{\theta_i} - \mu_i^{15} \tau_{\kappa_i} + 2\mu_i^{15} - \mu_i^{15} \overline{T_{\theta_i,\xi}} = 0$
- $-\mu_i^{14} + D_{\theta_i,\xi}(-\mu_i^{15} \tau_{\theta_i} - \mu_i^{15} \tau_{\kappa_i} + 2\mu_i^{15} - \mu_i^{15} \overline{T_{\theta_i,\xi}}) = 0$
- $D_{\theta_i,\xi} = \frac{\mu^{14}}{-\mu_i^{15} \tau_{\theta_i} - \mu_i^{15} \tau_{\kappa_i} + 2\mu_i^{15} - \mu_i^{15} \overline{T_{\theta_i,\xi}}}$
- **if** $\mu^{14} \geq 0$ **then**
  - Z$_{sub19} = convex$
  - **if** $D_{\theta_i,\xi}$ *is feasible* **then**
    - Dtheta[i] = $D_{\theta_i,\xi}$
  - **else**
    - set Dtheta[i] to min(V_1, V_2), which is the boundary value with smaller objective value
  - **end**
- **else**
  - set Dtheta[i] to min(V_1, V_2), which is the boundary value with smaller objective value
- **end**

end

doi:10.6342/NTU202210101

4.4.2.20 Subproblem 20(related to decision variable $q_{\kappa S_i}$)

By extracting all decision variables $q_{\kappa S_i}$ in the LR problem, we will then obtain subproblem 20. In subproblem 20, there are 3 multipliers μ_i^2,μ_i^{11} and μ^{13} and all of them can be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity and the solution to subproblem 20 is similar to subproblem 7. We have to take the sign of μ_i^2,μ_i^{11} and μ^{13}into consideration, but not μ^{11} because $q_{\kappa S_i}$ is linear so it won't affect the concavity or convexity of subproblem 20.

We separate $q_{\kappa S_i}$ from the objective function and derive a subproblem shown as formula (4.233) and constraint shown as formula (4.234).

Table 4.41: Subproblem 20(related to decision variable $q_{\kappa S_i}$)

Objective function :

$$Z_{sub20} = \min \sum_{i \in N} (\mu_i^2 \log q_{\kappa S_i} + \mu_i^{11} q_{\kappa S_i} + \mu^{13} \log q_{\kappa S_i}) \tag{4.233}$$

Subject to :

$$\epsilon \leq q_{\kappa S_i} \leq 1 \qquad \forall i \in N \tag{4.234}$$

doi:10.6342/NTU202210101

Algorithm 41: Algorithm for Subproblem 20

for *each node i* **do**
  Calculate the objective value at $q_{\kappa S_i} = \epsilon$ denoted as V_1
  Calculate the objective value at $q_{\kappa S_i} = 1$ denoted as V_2
  partial differential to $q_{\kappa S_i}$
  $\mu_i^2 \frac{1}{q_{\kappa S_i} \ln e} + \mu_i^{11} \frac{1}{q_{\kappa S_i} \ln e} + \mu^{13} = 0$
  $\frac{\mu_i^2 + \mu_i^{11}}{q_{\kappa S_i}} = -\mu^{13}$
  $q_{\kappa S_i} = -\frac{\mu_i^2 + \mu_i^{11}}{\mu^{13}}$
  if $Z_{sub8}(q_{\kappa S_i}) \leq \min(V_1, V_2)$ **then**
    $\mathrm{Z}_{sub20} = convex$
    if $q_{\kappa S_i}$ *is feasible* **then**
      qkappa[i] = $q_{\kappa S_i}$
    else
      set qkappa[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
    end
  else
    set qkappa[i] to $\min(V_1, V_2)$, which is the boundary value with smaller objective value
  end
end

4.4.2.21 Subproblem 21(related to decision variable $I_{\kappa S_i}$)

By extracting all decision variables $I_{\kappa S_i}$ in the LR problem, we will then obtain subproblem 21. In subproblem 21, there exists multiplier μ_i^2 and μ_i^{11}. μ_i^2 and μ_i^{11} can both be either positive or negative due to relaxing equality constraints.

The way to determine the concavity or convexity of subproblem 21 and the solution is similar to subproblem 1. We separate $I_{\kappa S_i}$ from the objective function and derive a subproblem shown as formula (4.235) and constraint shown as formula (4.236) .

doi:10.6342/NTU202210101

Table 4.42: Subproblem 21(related to decision variable $I_{\kappa R_i}$)

Objective function : $$Z_{sub21} = \min \sum_{i \in N} (\mu_i^2 (n-1) \log I_{\kappa S_i} + \mu_i^{11} I_{\kappa S_i}) \quad (4.235)$$ **Subject to :** $$0 \leq I_{\kappa S_i} \leq 1 - \epsilon \qquad \forall i \in N \quad (4.236)$$

Algorithm 42: Algorithm for Subproblem 21

```
for each node i do
    Calculate the objective value at I_{κS_i} = 0 denoted as V_1
    Calculate the objective value at I_{κS_i} = 1 − ϵ denoted as V_2
    partial differential to I_{κS_i}
    μ_i^2 (n−1)/(I_{κS_i} ln e) + μ_i^{11} = 0
    I_{κS_i} = −μ_i^2(n−1)/μ_i^{11}
    if Z_{sub21}(I_{κS_i}) ≤ min(V_1, V_2) then
        Z_{sub21} = convex
        if I_{κS_i} is feasible then
            IkappaS[i] = I_{κS_i}
        else
            set IkappaS[i] to min(V_1, V_2), which is the boundary value with
              smaller objective value
        end
    else
        set IkappaS[i] to min(V_1, V_2), which is the boundary value with smaller
          objective value
    end
end
```

4.5 Lagrangian Dual Problem and The Subgradient Method

In this minimization problem, the main purpose of the LR problem (Z_{LR}) is to provide us a bound to the optimal solution. By keep decreasing the gap between the upper bound and lower bound, we can obtain a better solution throughout the process. The optimal

doi:10.6342/NTU202210101

solutions of the subproblems will form the optimal value of the LR problem, which is the lower bound of the primal problem. And the feasible solutions to the primal problem will form an upper bound.

The lower bound have to keep increase for the purpose of to approach the optimal solution of the primal problem. In order to achieve that, a dual problem (Z_D) will be formed. The aim of the dual problem is to maximize the objective value of the LR problem so as to find the tightest lower bound. The dual problem is subjected to the Lagrangian multipliers (μ), and by adjusting multipliers, we can achieve smaller duality gap to find the maximum value of the dual problem. Table (4.43),(4.44) and (4.45) respectively shows the dual problem of Model 1 (One-to-one relationship), Model 2 (Many-to-one relationship) and Model 3 (Network Tree Structure) .

Table 4.43: The dual problem of Model 1(One-to-one relationship)

Objective function :

$$Z_D = \max Z_{LR}(\mu^1, \mu^2, \mu^3, \mu^4, \mu^5, \mu^6, \mu^7) \tag{4.237}$$

Table 4.44: The dual problem of Model 2(Many-to-one relationship)

Objective function :

$$Z_D = \max Z_{LR}(\mu^1, \mu^2, \mu^3, \mu^4, \mu^5, \mu^6, \mu^7) \tag{4.238}$$

doi:10.6342/NTU202210101

Table 4.45: The dual problem of Model 3(Network Tree Structure)

Objective function :

$$Z_D = \max Z_{LR}(\mu^1, \mu^2, \mu^3, \mu^4, \mu^5, \mu^6, \mu^7, \mu^8, \mu^9, \mu^{10}, \mu^{11}, \mu^{12}, \mu^{13}, \mu^{14}, \mu^{15}, \mu^{16}) \tag{4.239}$$

Subject to :

$$\mu^{15}, \mu^{16} \geq 0 \tag{4.240}$$

When solving non-differential optimization problem, there are two conditions to make sure the subgradient method is guaranteed to optimally solve the problem [43]. The stepsize in the subgradient method should converge to 0, but not too quickly or else the subgradient method will converge to a point other than the optimal solution. The second condition is the sum of the stepsize sequence should be infinity, which means the sequence of the stepsize should be a divergent series. These two conditions are shown as formula (4.241) below where t^k is the stepsize parameter and k is the iteration count.

When $k \to \infty$:

$$t^k \to 0 \qquad and \qquad \sum_{i=1}^{k} t^i \to \infty \tag{4.241}$$

We use The Subgradient Method to solve the dual problem in this research. The Subgradient Method is proposed by Held and Karp [45] [46] and is a effective way to solve the Lagrangian dual problem. The Lagrangian multipliers are updated by the formula (4.242) shown as below.

doi:10.6342/NTU202210101

$$\mu^{k+1} = \max(0, \mu^k + t^k g^k) \tag{4.242}$$

μ_k is a vector that contains all the multipliers, and t_k is used to determine the stepsize in order to update the multipliers. g_k is the subgradient of the dual problem. We use formula (4.243) shown as below to determine the stepsize parameter t_k in this research.

$$t^k = \frac{\lambda_k(Z^* - Z_D(\mu^k))}{||g^k||} \tag{4.243}$$

In formula (4.243), Z^* is the objective value of the best known feasible solution that we have obtained so far. λ_k is the coefficient of the stepsize which is often initially step to 2. When $Z_D(\mu^k)$ has failed to increase in a certain number of iteration that we set, λ_k will be reduced by a factor of two. The objective value of the feasible solution Z^* sometimes can be initially set to 0 and can be updated when solutions to the LR problem turns out to be feasible, or we can tune the solutions that aren't feasible by proposed heuristics.

As the number of adjustment of the multipliers increases, the duality gap between the lower bound and the upper bound will gradually reduce. There are few criteria for the solution process to terminate. The first one is when we obtain a μ^k for which $Z_D(\mu^k)$ (LB) overlaps Z^*(UB), and then it means the problem is optimally solved. The second one is when the duality gap size is less than the threshold value we set and a fairly high-quality solution will be obtained. The last one is when the number of iterations reached the limit we set. The pseudo code (**Algorithm** 43) of the Lagrangian Relaxation method with The Subgradient Method is shown as below.

doi:10.6342/NTU202210101

Algorithm 43: Lagrangian Relaxation method with The Subgradient Method

Input : N : max number of iterations;
$target\ gap$: target optimization gap;

Initialization: x^0 : any feasible solution;
$Z^*(x^k)$: primal feasible solution value for x
Z_{LB}^0 : $-inf$;
Z_{UB}^0 : best known feasible solution for objective function, objective value for x^0
μ : initial multiplier value = 0
$current\ best\ solution$: x^0
$current\ best\ objective\ value$: Z_{UB}^0
k : iteration count = 0
i : improvement count = 0
λ : initial step size coefficient = 2

for $k \leq N$ **do**
  Solve Z_{LR} :
    Get Solution x^k ;
    Compute current objective value $Z_D(\mu_k, x^k)$;
  Compute subgradient $g_k = g(x^k)$;
  if x^k $is\ feasible$ **then**
    $Z^*(x^k)$ $is\ an\ upper\ bound$
  else
    tune x^k $with\ proposed\ heuristic$
  end
  $Z_{LB}^{k+1} = \max(Z_{LB}^k, Z_D(\mu^k, x^k))$;
  if $Z^* \leq Z_{UB}^k$ **then**
    $current\ best\ solution = x^k$
    $current\ best\ objective\ value = Z^*(x^k)$
  end
  $Z_{UB}^{k+1} = \min(Z_{UB}^k, Z^*)$
  if $Z_D(\mu^k, x^k) \leq Z_{LB}^k$ **then**
    $i = i + 1$
  end
  $gap^{k+1} = \frac{Z_{UB}^{k+1} - Z_{LB}^{k+1}}{Z_{LB}^{k+1}}$
  if *i reaches improvement counter limit* **then**
    $\lambda = \lambda/2$
    i = 0
  end
  Compute t^k *(step size of subgradient descent)*
  $t^k = \frac{\lambda_k(Z^*(x^k) - Z_D(\mu^k, x^k))}{||g_k||^2}$
  $\mu^{k+1} = \max(0, \mu^k + t^k g^k)$
  if $gap^{k+1} \leq target\ gap$ **then**
    return $current\ best\ solution$, $current\ best\ objective\ value$, gap^{k+1}
  end
  end
  return $current\ best\ solution$, $current\ best\ objective\ value$, gap^{k+1}

doi:10.6342/NTU202210101

4.6 Getting Primal Feasible Solution

After solving each subproblems that were formed, we then obtain set of decision variables. The next step is to check whether the decision variables are feasible. If so, then an UB is formed by the objective value of the primal problem. However, if the decision variables were not feasible, heuristic methods would become crucial in order to tune the decision variables into feasible solutions. In this study, we propose different getting primal feasible solutions for each models due to their characteristics and features.

4.6.1 Model 1(One-to-one relationship)

In model 1, we first take this constraint(4.244) into consideration.

$$\tau_i \left[\frac{1}{q_i q_j P_{ij}(r_i, m)} - 1 \right] + 1 \leq \overline{T_{ij}} \qquad \forall i \in S, j \in S, m \in M \tag{4.244}$$

If we are going to find the initial primal feasible solution for the primal problem, the easiest way is to set all q_i, q_j and r_i to the max. That way the time slots needed to transmit a single packet will be minimized. In this model we assume that whenever a node is active it will send data, so in such way the delay constraint(4.244) will be satisfied even though it will consume more energy.

After the constraint is satisfied, we can adjust each decision variables to minimize the energy consumption and at the same time satisfy the constraint. Lagrangian multipliers

doi:10.6342/NTU202210101

can play an important role for distinguishing which decision variable is more significant to the model and should have the higher priority to adjustment. Further and detailed studies about the use of Lagrangian multipliers will be provided and demonstrated in the next section.

4.6.2 Model 2(Many-to-one relationship)

In model 2, the method to get the primal feasible solution is more complicated than model 1. We take constraint(4.245) into consideration.

$$\tau_i \left[\frac{1}{q_i(1-q_i)^{n-1} q_\kappa P_{i\kappa}(r_i, m)} - 1 \right] + 1 \leq \overline{T_{i\kappa}} \qquad \forall i \in S, m \in M \qquad (4.245)$$

There are three variables q_i, q_κ and r_i to be adjust. The time slots before the first successful transmission appears when $q_i = \frac{1}{n}$, q_κ and $P_{i\kappa}(r_i, m) = 1$.

As for q_i, there exists no advantage when q_i is greater than $\frac{1}{n}$ for both the delay constraint(4.245) or the minimization of the primal problem. When q_i is greater than $\frac{1}{n}$, the time slots needed before the first successful transmission will increase and may become infeasible. Also when q_i is greater than $\frac{1}{n}$, it will increase the energy consumption of the primal problem.

Therefore, when the decision variables we obtain is infeasible, we set q_i to $\frac{1}{n}$, q_κ and $P_{i\kappa}(r_i, m)$ to 1. However, when q_i is smaller than $\frac{1}{n}$, it might reduce the power consumption. Hence, when the decision variables are satisfied, we can adjust them respectively so

doi:10.6342/NTU202210101

as the constraint won't be break and the power consumption can be minimized.

Lagrangian multipliers can play an important role for distinguishing which decision variable is more significant to the model and should have the higher priority to adjustment. It can be used as the sensibility or the penalty coefficient of the model. It can show some hints on whether a constraint or a variable is important for the model. With this kind of hint, we can decide which decision variable to adjust first so it will benefit the model most.

Decision variable q_i and r_i exists different multipliers for each i, and q_κ exist a single multiplier. In the next chapter we will discuss and analyze whether using the summation, the largest or the average of the multipliers in the set i is the most appropriate and effective.

4.6.3 Model 3(Network Tree Structure)

In model 3, not only we have to consider the delay constraint(4.246), we also have to consider the newly added throughput constraint(4.247) as shown as followed.

$$\tau_{\theta_i}\left[\frac{1}{q_{\theta_i}(1-q_{\theta_i})^{(v-1)}q_{\kappa R_i}P_{\theta_i,\kappa_i}(r_{\theta_i},m)}-1\right]+\tau_{\kappa_i}\left[\frac{1}{q_{\kappa S_i}(1-q_{\kappa S_i})^{(n-1)}q_{\xi}P_{\kappa_i,\xi}(r_{\kappa_i},m)}-1\right]+2\leq\overline{T_{\theta_i,\xi}}$$

$$\forall i\in N, m\in M \tag{4.246}$$

$$mq_{\kappa S_i}(1-q_{\kappa S_i})^{(n-1)}q_{\xi}P_{\kappa_i,\xi}(r_{\kappa_i},m)\geq mvq_{\theta_i}(1-q_{\theta_i})^{(v-1)}q_{\kappa R_i}P_{\theta_i,\kappa_i}(r_{\theta_i},m)$$

$$\forall i\in N, m\in M \tag{4.247}$$

As for the delay constraint, more factors and decision variables appears compared to model 1 and model 2. And in model 3, we assume that the relay nodes κ have two

doi:10.6342/NTU202210101

subsystems. One in charge of the activeness of receiving ($q_{\kappa R_i}$) and another one is in charge of the activeness of sending ($q_{\kappa S_i}$).

We first analyze the delay constraint(4.246) and it appears that when $q_{\theta_i} = \frac{1}{v}$, $q_{\kappa S_i} = \frac{1}{n}$, $q_{\kappa R_i}$, $P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$, q_ξ and $P_{\kappa_i,\xi}(r_{\kappa_i}, m) = 1$, the time slots needed for a successful transmission is minimized. It is similar with model 1 and model 2.

On the other hand, when the throughput constraint(4.247) is violated, setting $q_{\kappa S_i}$ to $\frac{1}{n}$ and q_ξ, $P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ to 1 is the first step to tune the decision variables. It will make the left side of the throughput constraint(4.247) as big as possible. The reason why we choose to first tune $q_{\kappa S_i}$, q_ξ and $P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ is that tuning these decision variables is more effective than tuning q_{θ_i}, $q_{\kappa R_i}$ and $P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ because by tuning $q_{\kappa S_i}$ to $\frac{1}{n}$ and q_ξ, $P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ to 1, it not only helps obey the delay constraint(4.246) but also the throughput constraint(4.247).

After that, we check the delay constraint(4.246) whether it is violated or not. If so, we set q_{θ_i} to $\frac{1}{v}$ and $q_{\kappa R_i}$, $P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ to 1 and then gradually decrease them so as both the delay constraint(4.246) and the throughput constraint(4.247) can be satisfied. From the perspective of throughput, q_{θ_i}, $q_{\kappa R_i}$ and $P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ will be forced to decrease to satisfy the throughput constraint(4.247) in order to avoid the problem of overflowing.

At last when both constraints are satisfied, we gradually decrease $q_{\kappa S_i}$, q_ξ and r_{κ_i} to get a better objective value to the primal problem and still remain the feasibility of the

doi:10.6342/NTU202210101

solution.

doi:10.6342/NTU202210101

Chapter 5 Experimental Results and Discussion

In this chapter, several experiments regarding wireless sensor network are conducted to verify the validity and the performance of our proposed method. The main goal of this research is to minimize the energy consumption of the wireless sensor network in different scenarios. Hence, results of different experiment cases and discussions will be presented in the following section.

5.1 Experimental Environment

The experiments conducted in this research are implement by Python language on Jupyter notebook IDE with version 3.8.5. We use a desktop as the execution environment. The system parameters are shown in detailed in Table 5.1. The parameters related to the LR method are shown in Table 5.2.

doi:10.6342/NTU202210101

Table 5.1: System Parameters

Parameters	Value
Computer Type	Desktop
Central Processing Unit	Intel Core i5-6400 2.70 GHz
System Type	x64-based pc
Random Access Memory	32GB
Programming Language	Python
Integrated Development Environment	Jupyter notebook

Table 5.2: Lagrangian Relaxation Parameters

Parameters	Value
Iteration Count Limit	300
Improvement Count Limit	10
Initial Lambda	2
Initial Multipliers	0.00001
Initial UB	∞
Initial LB	$-\infty$

doi:10.6342/NTU202210101

5.2 Experiments and Results

5.2.1 The Design of Probability and Energy Consumption Functions

$P_{i\kappa}(r_i, m)$, $P_{\theta_i,\kappa_i}(r_{\theta i}, m)$ and $P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ (P) are the probability functions used to transmit a packet with $m \in M$ size with transmission range radius of $r_i \in R_i$, $r_{\theta i} \in R_{\theta i}$ and $r_{\kappa_i} \in R_{\kappa_i}$. We take $P_{i\kappa}(r_i, m)$ as an example. When we set the transmission range radius r_i to fixed, the higher the packet size m is, the lower the probability of transmitting a packet is. However, when we set the packet size m to fixed, the higher the transmission range radius r_i is , the stronger the signal, the higher the Signal-to-noise ratio(SNR) rate and therefore the higher the probability of transmitting a packet is. And in our design, function P is a concave function which will saturated or asymptotically converge to 1 .

In order to implement this function P, we employ the energy consumption characteristics of Mica2 motes [47] equipped with CC1000 [48]. Mica2 motes is the most commonly utilized sensor nodes in experimental WSN research due to their well-characterized energy dissipation properties [33] and CC1000 is a RF Module, which is a small electronic device used to transmit and/or receive radio signals between two devices.

The transmission power consumption (P_{tx}^{crc}) and output antenna power (P_{tx}^{ant}) at each power level(l)[48] is shown as Figure 5.1.

doi:10.6342/NTU202210101

l	$P_{tx}^{crc}(l)$	$P_{tx}^{ant}(l)$	l	$P_{tx}^{crc}(l)$	$P_{tx}^{ant}(l)$
1 (l_{min})	25.8	0.0100	14	32.4	0.1995
2	26.4	0.0126	15	33.3	0.2512
3	27.0	0.0158	16	41.4	0.3162
4	27.1	0.0200	17	43.5	0.3981
5	27.3	0.0251	18	43.6	0.5012
6	27.8	0.0316	19	45.3	0.6310
7	27.9	0.0398	20	47.4	0.7943
8	28.5	0.0501	21	50.4	1.0000
9	29.1	0.0631	22	51.6	1.2589
10	29.7	0.0794	23	55.5	1.5849
11	30.3	0.1000	24	57.6	1.9953
12	31.2	0.1259	25	63.9	2.5119
13	31.8	0.1585	26 (l_{max})	76.2	3.1623

Figure 5.1: Power Consumption of Mica2 motes

The received signal power ($Po_{rx,ij}^{ant}$) due to a transmission at power level-l over link-(i, j) is shown as formula(5.1).

$$Po_{rx,ij}^{ant}(l) = P_{tx}^{ant}(l) - \gamma_{ij} \tag{5.1}$$

$$\gamma_{ij} = \gamma_0 + 10a\log(\frac{d_{ij}}{d_0}) + X_\sigma \tag{5.2}$$

Where γ_{ij} is the path loss value occurred over link-(i, j), based on the log-normal shadowing path loss model [49]. d_{ij} is the distance between transmitter and receiver, γ_0 is the path loss at the reference distance, a is the path loss exponent and X_σ is a zero-mean Gaussian random variable with the standard deviation σ dB to model large-scale fading (shadowing) effects. We adopt the parameter values provided for Mica2 motes as n = 3.69, σ = 1.42 dB, d0 = 1 m, and Y0 = 31 dB [50].

SNR is defined as

doi:10.6342/NTU202210101

$$\psi_{ij}(l) = Po^{ant}_{rx,ij}(l) - Po_n \tag{5.3}$$

The noise power (Po_n) is −115 dBm at the temperature of 300 Kelvin for Mica2 motes [51]. The probability of a successful packet reception of a m-Byte packet transmitted at power level-l over the link-(i, j) [51] is shown as formula(5.4)

$$p^s_{ij}(l, m) = \left(1 - \frac{1}{2}\exp\left(\frac{-\psi_{ij}(l)}{2}\frac{1}{0.64}\right)\right)^{8m} \tag{5.4}$$

The distance over the link-(i, j) versus the probability of successfully transmitting a packet is shown as Figure 5.2.

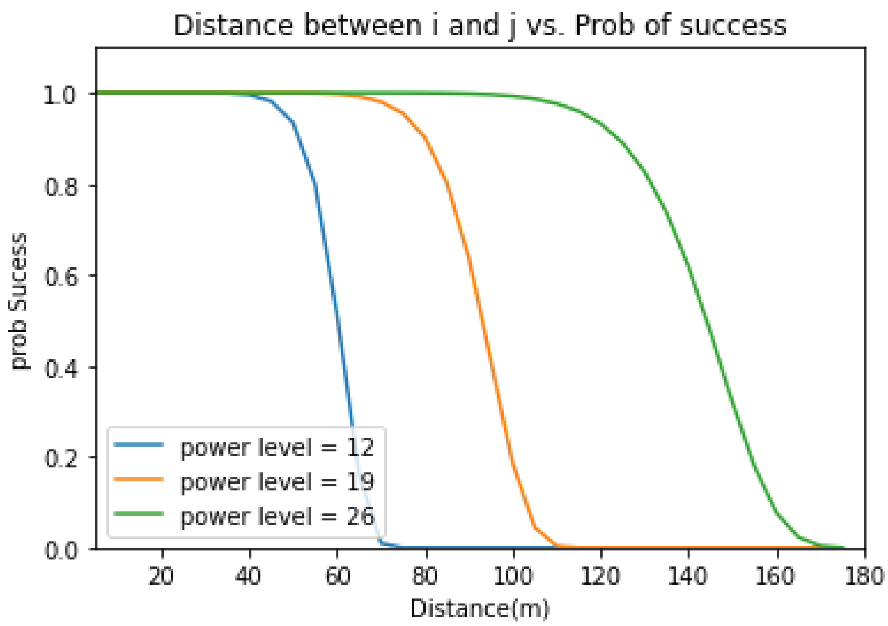

Figure 5.2: Distance between i and j vs. Prob of success

We then implement the probability function P with the characteristics of Mica mote2 and replace power level-l with transmission range radius r_i. The figure of the probability of node i to transmit packet with m size when no error occurs with transmission range radius of r_i is shown as Figure 5.3.

doi:10.6342/NTU202210101

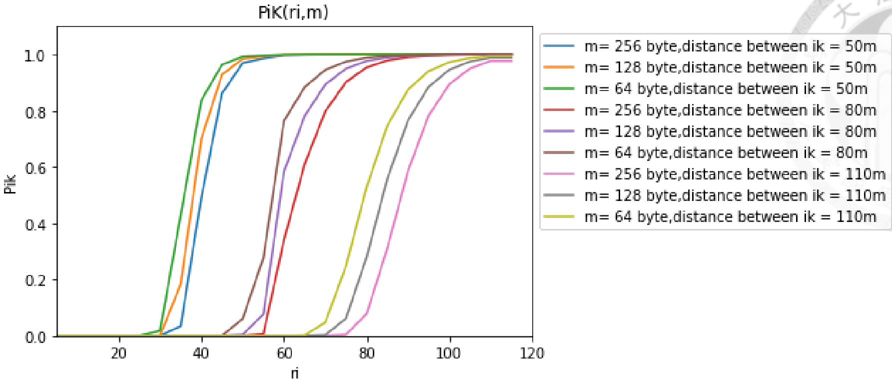

Figure 5.3: Probability function

As for energy consumption functions $C^a(ri, m)$, $C^b(ri, m)$ and $C^\tau(ri, m)$, we implement them respectively with the power consumption for reception of Mica mote 2[23], the power consumption in the sleep mode and the power consumption shown as Figure 5.1. The relationship of power consumption and the transmission range radius of r_i is shown as Figure 5.4.

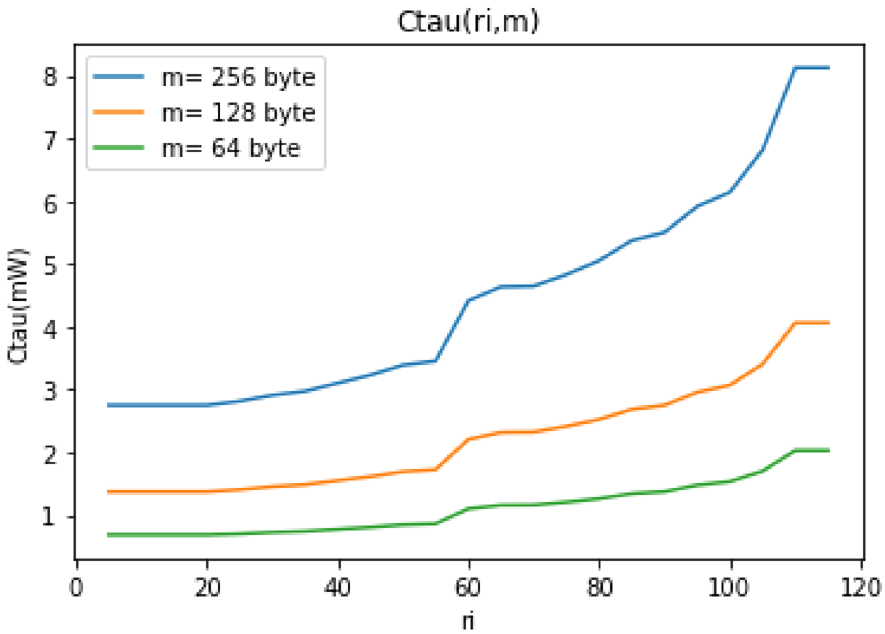

Figure 5.4: C^τ

doi:10.6342/NTU202210101

5.2.2 Experiment 1

In experiment 1, we conducted an experiment to testify the performance of our proposed method introduced in Many-to-One model. We set the distance between sensor nodes and relay node to 30 meters. The number of sensor nodes is set to 10. The timeout interval τ_i is set to 10 time slots. In terms of timeout interval, time and the factors of environment will influence it, but we assume that our environment is a quasistatic environment. In such way we can assume that the timeout interval τ_i is fixed. When a packet is transmitted, if the expected acknowledgement isn't received before τ_i time slots, it is considered a failure transmission. It is composed of a packet transmission delay, propagation delay, receiver's processing delay, the time needed for forming the acknowledgement, the transmission delay for the acknowledgement, the propagation delay for the acknowledgement and the processing delay for the source node to receive the acknowledgement. We assume that each of them need 1 time slot, therefore it needs at least 7 time slots to proceed the mentioned process.

The setting of $T_{i\kappa}$ is crucial. When $T_{i\kappa}$ is set too small, sensor nodes will not be able to transmit data in time. If $T_{i\kappa}$ is set too big, the delay constraint will be considered as useless. We first observe and analyze the relationship between the number of sensor(n) nodes and $T_{i\kappa}$. We take delay constraint(5.5) below into consideration. The least amount of delay $T_{i\kappa} = 249.37585528$ happens when $q_i = \frac{1}{n}, q_\kappa = 1$ and $P_{i\kappa} = 1$.

doi:10.6342/NTU202210101

$$\tau_i\left[\frac{1}{q_i(1-q_i)^{n-1}q_\kappa P_{i\kappa}(r_i,m)}-1\right]+1\leq\overline{T_{i\kappa}}\qquad\forall i\in S, m\in M\tag{5.5}$$

Let $q_i=\frac{1}{n}, q_\kappa=1, P_{i\kappa}=1$:

$$\tau_i\left[\frac{1}{q_i(1-q_i)^{n-1}q_\kappa P_{i\kappa}(r_i,m)}-1\right]+1=\tau_i\left[\frac{1}{\frac{1}{n}(1-\frac{1}{n})^{n-1}}-1\right]+1\tag{5.6}$$

When $n\to\infty$:

$$\lim_{n\to\infty}\tau_i\left[\frac{1}{\frac{1}{n}(1-\frac{1}{n})^{n-1}}-1\right]+1\qquad\leq\overline{T_{i\kappa}}\tag{5.7}$$

$$=\lim_{n\to\infty}\tau_i\left[\frac{1}{\frac{1}{n}e^{-1}}-1\right]+1\qquad\leq\overline{T_{i\kappa}}\tag{5.8}$$

$$=\lim_{n\to\infty}\tau_i\left[n\times e-1\right]+1\qquad\leq\overline{T_{i\kappa}}\tag{5.9}$$

From the equations above, we can find out that the number of sensor(n) nodes is proportional to $T_{i\kappa}$. We then set $T_{i\kappa}$ to 260, which is considered extremely tight in order to make the feasible region very small and test if our method can find the expected answer for the decision variables.The result is shown as Figure 5.5.

From Figure 5.5, we can see that the getting primal feasible solution gets the optimal solution throughout the first few iterations. When iteration continues to increase, the lower bound gradually approaches the upper bound causing the gap between them to shrink. After 100 iterations, the gap between the lower bound and upper bound is 4.7 % and the time spent is 20.44 seconds. It is shown as Table 5.3.

doi:10.6342/NTU202210101

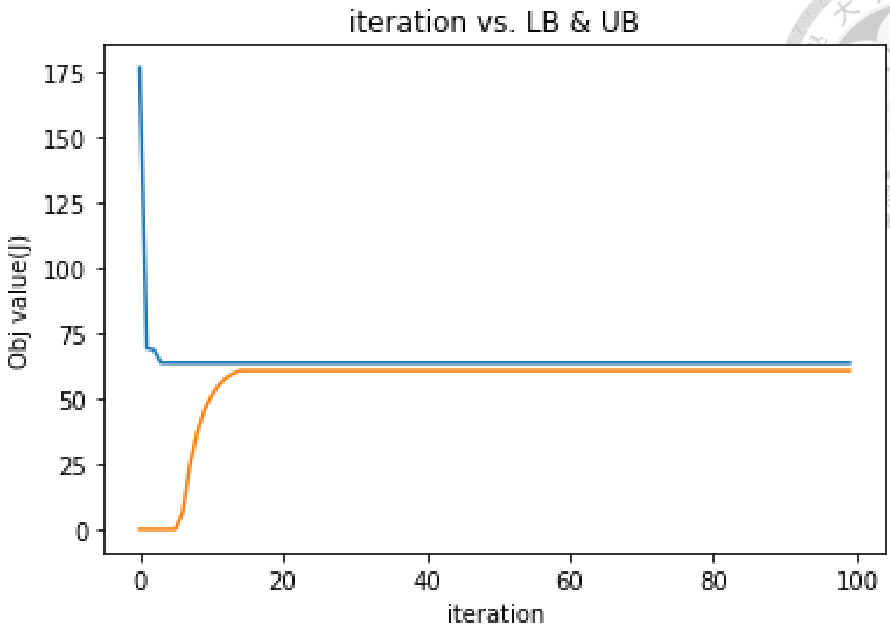

Figure 5.5: Many-to-One LR Result

Table 5.3: Many-to-One LR Result

LB	**UB**	**Gap**	**Time**
60.56	63.45	4.7 %	20.4

5.2.3 Experiment 2

In experiment 2, we set the delay constraint $T_{i\kappa}$ even tighter (249.37585529) to conduct an experiment to testify the performance of our proposed method. We set the distance between nodes to 30 meters. The timeout interval τ_i is set to 10 time slots as explained in the previous section. The number of sensor nodes is set to 10.The result is shown as Figure 5.6 and Table 5.4 .

doi:10.6342/NTU202210101

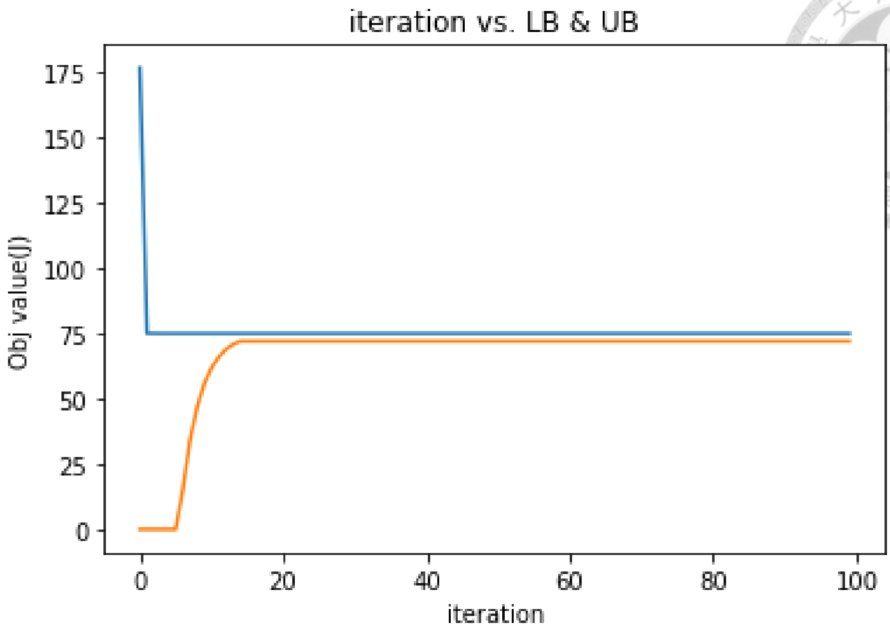

Figure 5.6: Many-to-One LR Result 2

Table 5.4: Many-to-One LR Result 2

LB	**UB**	**Gap**	**Time**
71.9	74.82	4.08 %	15.8

We then also conduct an experiment when sensor nodes is set to 5, and set the delay constraint $T_{i\kappa}$ very tight (113.1925051) to conduct the experiment. The number of sensor nodes is set to 10.The result is shown as Figure 5.7 and Table 5.5 .

doi:10.6342/NTU202210101

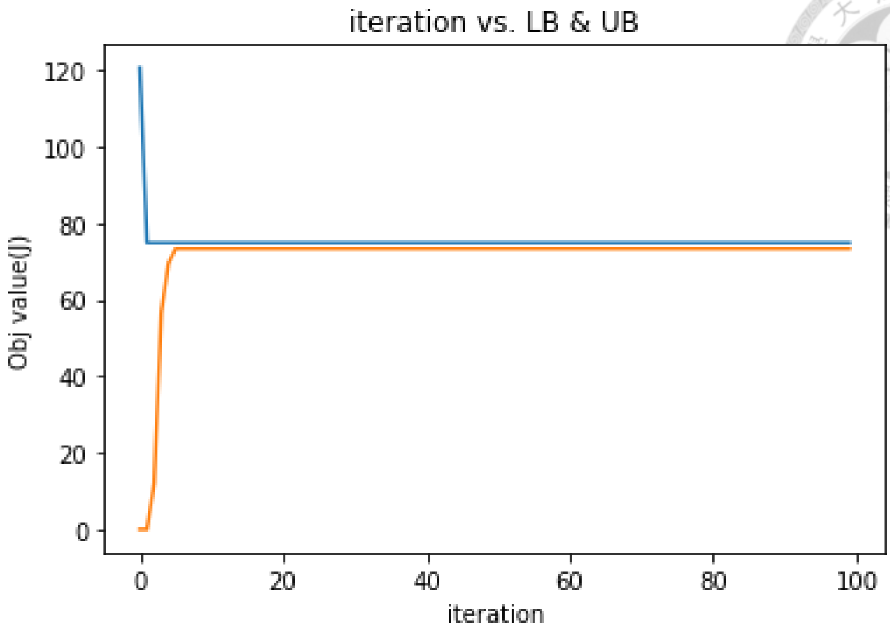

Figure 5.7: Many-to-One LR Result 3

Table 5.5: Many-to-One LR Result 3

LB	**UB**	**Gap**	**Time**
73.3	74.84	2.04 %	4.8

From the experiments shown above, we can find out that the probability of the sensor node to be active when n = 5 is two times as large as when n = 10. The reason is that when the delay constraint is extremely strict, in order to get the optimal solution, the probability of the sensor node to be active will approach to $\frac{1}{n}$. The probability of the relay node to be active and the probability to transmit a packet without error will approach to 1. Therefore, the total power consumption when n = 5 is similar to n = 10 because the number of nodes when n = 10 is two times as large as when n = 5, and the probability of the sensor node to be active when n = 5 is two times as large as when n = 10.

doi:10.6342/NTU202210101

5.2.4 Experiment 3

In experiment 3, we conducted an experiment to testify the performance of our proposed method introduced in the Network Tree Structure model. We set the distance between nodes to 30 meters. The number of sensor nodes in each subtree(v) is set to 2 and the number of subtree(n) is set to 5. The timeout interval τ_i is set to 10 time slots as explained in the previous section. $T_{\theta_i,\xi}$ is set to 1001 which is also tight to testify the performance of our proposed model. The result is shown as Figure 5.8 and Table 5.6 .

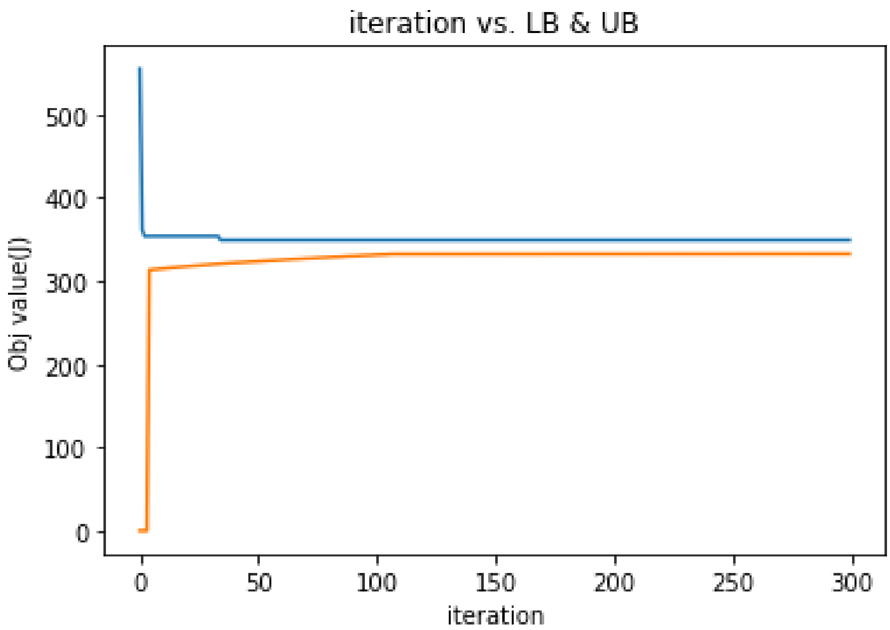

Figure 5.8: Network Tree Structure Experiment 1

Table 5.6: Network Tree Structure Experiment 1

LB	UB	Gap	Time
332.75	349.38	4.9 %	308.6

doi:10.6342/NTU202210101

We then increase the number of sensor nodes in each subtree(v) into 3 and the number of subtree(n) into 8 and do the experiment in a bigger scale. $T_{\theta_i,\xi}$ is set to 3000. The result is shown as Figure 5.9 and Table 5.7 .

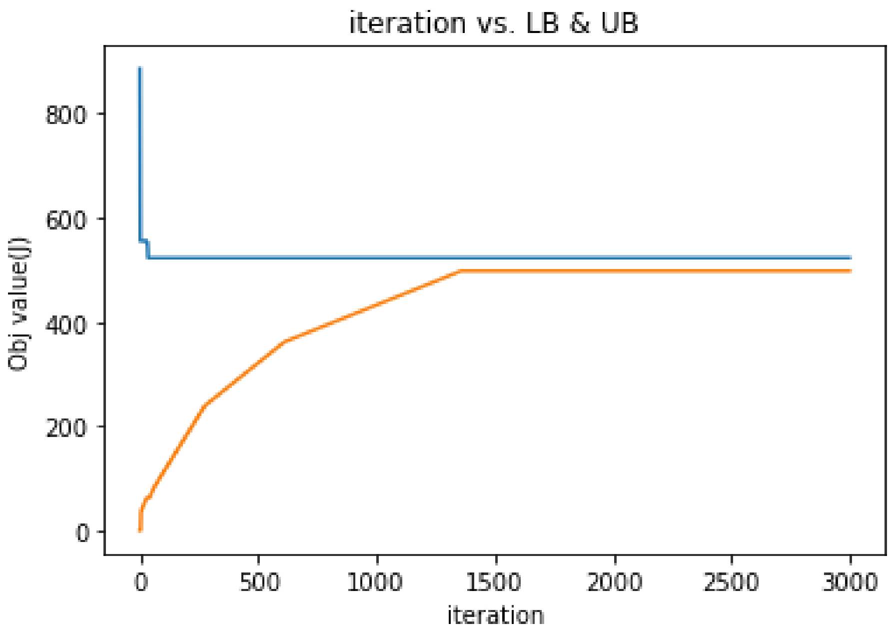

Figure 5.9: Network Tree Structure Experiment 2

Table 5.7: Network Tree Structure Experiment 2

LB	**UB**	**Gap**	**Time**
498.06	522.76	4.96 %	4662.7

From the results shown above, we found out that at the beginning, the model will tend to adjust $q_{\kappa S_i}$, q_ξ and $P_{\kappa_i,\xi}(r_{\kappa_i}, m)$ as much as possible to satisfy both the delay constraint and the throughput constraint. After that, if the throughput constraint is not satisfied, q_{θ_i}, $q_{\kappa R_i}$ and $P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ will be decreased according to the size of multipliers to make it feasible. At last when solutions are feasible, decision variables will be slightly adjust

doi:10.6342/NTU202210101

to try to find a better solution. The results also shows that compared to model 2, q_{θ_i}, $q_{\kappa R_i}$ and $P_{\theta_i,\kappa_i}(r_{\theta_i}, m)$ will be forced to set to a smaller value in order to satisfy the throughput constraint and to avoid overflow.

5.2.5 Experiment 4

In experiment 4, we conducted an experiment to observe the effect of different packet sizes(m). Packet sizes(m) is also considered as a decision variable. However, m and r_i often coupled together and is difficult to decompose. Therefore, we use exhausted solution to see which packet size m has the better solution.

There are three aspect that were influenced by m. First, the larger m is, the longer the time slot is for transmitting a packet. In this research, we use time slot as the unit, so we will also take the influence of packet size into consideration. Secondly, the probability of transmitting a packet without error will decrease when m increases. Thirdly, we assume each packet has a fixed size header of 50 bytes, therefore the larger the packet size is, the larger the throughput is. We take all the mentioned factors into consideration and conducted an experiment to see which m has the best performance. We use the same setting as experiment 2 and enumerate through 64 byte to 255 byte.

The result is shown as Figure 5.10 .

We can see that there exists a trade off. When m is too big, it could cause the prob-

doi:10.6342/NTU202210101

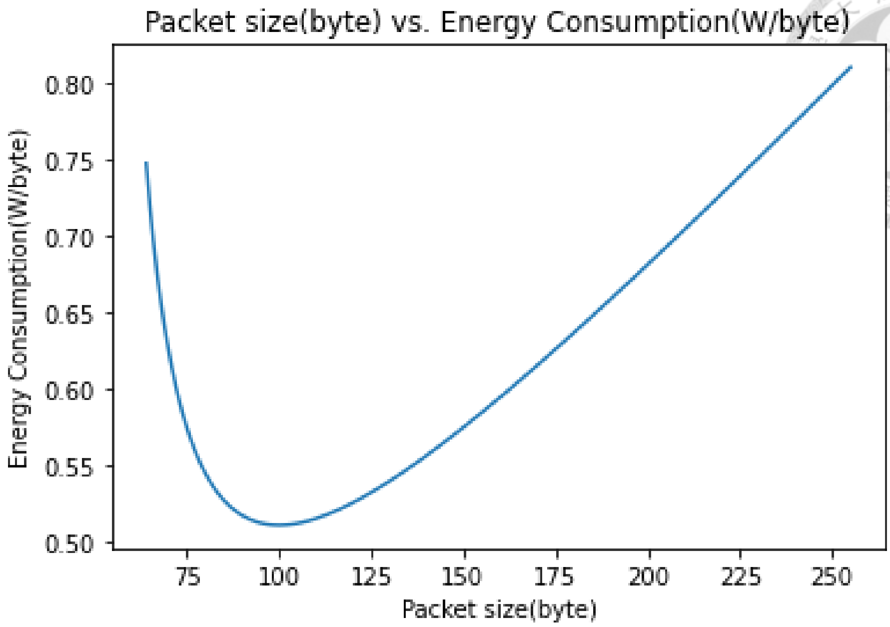

Figure 5.10: Packet Size Experiment

ability of transmitting a packet without error decrease, therefore the delay constraint will not be satisfied and also cause the absolute time interval of the time slot to increase and eventually increase the power consumption.

However, if m is too small, the real throughput will decrease. The reason is that the real throughput will be the packet size minus the header size. Therefore when m is too small, the power consumption of a single byte of the real throughput will be too high.

5.2.6 Experiment 5

In experiment 5, we conducted an experiment to observe and compare the usage of the multipliers in Many-to-One model. In the getting primal feasible solutions we proposed, we decide the adjustment order of the decision variables by comparing the multipliers. We

doi:10.6342/NTU202210101

compare using the multipliers by choosing the maximum in the set i, the summation of the set i and the average of the set i to see which way perform well in the getting primal feasible solution. We use the same setting as experiment 2 and enumerate through 64 byte to 255 byte. The result is shown as Table 5.8 .

Table 5.8: Multipliers Experiment 1

	Maximum	**Summation**	**Average**
Objective value	63.45	70.59	63.45

When comparing the usage of multipliers, we found out that by choosing the maximum multiplier in the set i and the average of the set i have the better performance than choosing the summation of the set i. It is because when the delay constraint is set tight, the multipliers of q_κ tend to become bigger than q_i, therefore the model tend to adjust q_κ first. However, when we use the summation of the set i, the model will then adjust q_i first. When the delay constraint is set tight, the adjustment of q_i will not be enough for satisfying it, so making q_i important by choosing the summation of the set i will lead to the worst performance. Also, choosing the maximum multiplier in the set i and the average of the set i does not affect the result because when choosing the the maximum multiplier in the set i, the multiplier is not big enough to influence the adjustment order of q_i, q_κ and r_i.

We then also set the adjustment order to fixed rather than using the multipliers to

doi:10.6342/NTU202210101

decide the adjustment order to see which perform better. There are three decision variables remain adjusting, so there will be 6 combinations of adjustment orders, which are [$q_i > q_\kappa > r_i$], [$q_i > r_i > q_\kappa$], [$q_\kappa > q_i > r_i$], [$q_\kappa > r_i > q_i$], [$r_i > q_i > q_\kappa$], [$r_i > q_\kappa > q_i$].The result is shown as Table 5.9 .

Table 5.9: Multipliers Experiment 2

	$q_i > q_\kappa > r_i$	$q_i > r_i > q_\kappa$	$q_i > q_\kappa > r_i$	$q_\kappa > r_i > q_i$	$r_i > q_i > q_\kappa$	$r_i > q_\kappa > q_i$
Objective value	70.258	70.229	64.913	63.828	70.029	63.456

We compared the above methods with the one using multipliers, and it shows that by choosing the adjustment order dynamically with multipliers has the best performance, which gets the smallest objective value. The result is shown as Figure 5.11 .

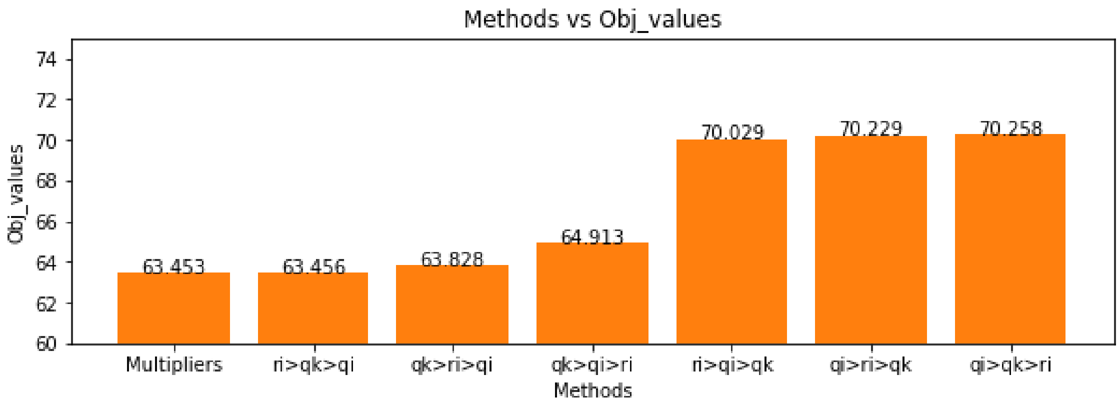

Figure 5.11: Multipliers Experiment 2

doi:10.6342/NTU202210101

5.2.7 Experiment 6

In experiment 6, we conducted an experiment to compare the performance and efficiency of our proposed method and Exhausted Search 1. We first conduct Exhausted Search 1 by searching all the possible answers of q_i and q_κ from 0.01 to 0.99. The result is shown as Table 5.10 .

Table 5.10: Exhausted Search 1 vs. Proposed Method

	Exhausted Search 1	**Proposed Method**
Objective value	64.458	63.453
Time	315.5	19.6

From the result above, we can see that the time that Exhausted Search 1 spent is ten times more longer than our proposed method. Moreover, the performance is worst than our proposed method. We then increase the precision of the Exhausted Search. We conducted Exhausted Search 2 by searching all the possible answers of q_i and q_κ from 0.001 to 0.999.The result is shown as Table 5.11 .

doi:10.6342/NTU202210101

Table 5.11: Exhausted Search 2 vs. Proposed Method

	Exhausted Search 2	**Proposed Method**
Objective value	63.189	63.453
Time	7527.8	19.6

From the result above, we can see that the performance of Exhausted Search 2 is better than our proposed method due to the precision increase. However, the time spent for Exhausted Search 2 is 300 times more than our proposed method.

doi:10.6342/NTU202210101

Chapter 6 Conclusions and Future Work

6.1 Conclusions

In this thesis, we proposed an optimization-based power control mechanism for saving energy in green wireless sensor networks. In order to minimize the energy consumption of the wireless sensor network and also satisfy the delay and throughput constraints, we developed a series of getting primal feasible solution and LR based algorithms. One-to-One model is first proposed theoretically and we later expand and implement it into Many-to-One model. At last, a complete and more practical Network Tree structure model is proposed by considering more scenarios and was implemented in a larger scale. After modeling, we apply LR method's divide-and-conquer characteristics to help us simplify the primal problem. By dividing the primal problems into subproblems, we can solve them individually in an easier way. We then conducted a series of experiments to testify

doi:10.6342/NTU202210101

the validity of our proposed method. The experiments shows different insights of saving energy in wireless sensor network and also demonstrates that the method we proposed performs better than other heuristics.

6.2 Future Work

There are some issues and works that could be done in the future to further improve this research.

In Network Tree structure model, we assume the relay node is consists of 2 subsystem, which can allow it to separately control the probability of activeness when receiving from the probability of activeness when sending. The connection between these 2 subsystems may exist some difficulties and extra energy consumption. After taking the above mentioned factor into consideration, we can then compare it with assuming that it only exist a single system responsible for controlling both the probability of activeness when receiving and the probability of activeness when sending. We presume that the model will be more flexible when we separate the relay node into 2 subsystem and the getting primal feasible solution will be able to adjust more easily to the throughput constraint. Therefore, research regarding the issues mentioned above could give us more insights and is worth deep discussion.

doi:10.6342/NTU202210101

References

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next century challenges: Scalable coordination in sensor networks,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 263–270, 1999.

[2] L. Zhaohua and G. Mingjun, “Survey on network lifetime research for wireless sensor networks,” in *2009 2nd IEEE International Conference on Broadband Network Multimedia Technology*, pp. 899–902, 2009.

[3] J. Vales-Alonso, E. Egea-López, A. Martínez-Sala, P. Pavón-Mariño, M. Victoria Bueno-Delgado, and J. García-Haro, “Performance evaluation of mac transmission power control in wireless sensor networks,” *Computer Networks*, vol. 51, no. 6, pp. 1483–1498, 2007.

[4] C. Rong, G. Zhao, L. Yan, E. Cayirci, and H. Cheng, “Chapter 10 - wireless network security,” in *Network and System Security (Second Edition)* (J. R. Vacca, ed.),

doi:10.6342/NTU202210101

pp. 291–317, Boston: Syngress, second edition ed., 2014.

[5] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[6] V. Mhatre and C. Rosenberg, "Design guidelines for wireless sensor networks: Communication, clustering and aggregation," *Ad Hoc Networks*, vol. 2, no. 1, pp. 45–63, 2004.

[7] C. Perkins, E. Royer, S. Das, and M. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 16–28, 2001.

[8] A. Djedouboum, A. Gueroui, A. Mohamadou, and Z. Aliouat, "Big data collection in large-scale wireless sensor networks," *Sensors*, vol. 18, 12 2018.

[9] V. P. Mhatre and C. P. Rosenberg, "Design guidelines for wireless sensor networks: Communication, clustering and aggregation," *Ad Hoc Networks*, vol. 2, pp. 45–63, 2004.

[10] M. O. Farooq and T. Kunz, "Wireless multimedia sensor networks testbeds and state-of-the-art hardware: A survey," in *Communication and Networking* (T.-h. Kim, H. Adeli, W.-c. Fang, T. Vasilakos, A. Stoica, C. Z. Patrikakis, G. Zhao, J. G. Vil-

doi:10.6342/NTU202210101

lalba, and Y. Xiao, eds.), (Berlin, Heidelberg), pp. 1–14, Springer Berlin Heidelberg, 2012.

[11] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of manet and wsn in iot urban scenarios," *Sensors Journal, IEEE*, vol. 13, pp. 3558–3567, 10 2013.

[12] D. Kumar, "Cloud communication and iot for smart devices in real time environment," *International Journal for Research in Applied Science and Engineering Technology*, vol. 6, pp. 1593–1597, 03 2018.

[13] G. Huang, "10 emerging technologies that will change your world," *Technology Review*, vol. 107, no. 1, pp. 32–+, 2004.

[14] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[15] S. P. Khapre, S. Chopra, A. Khan, P. Sharma, and A. Shankar, "Optimized routing method for wireless sensor networks based on improved ant colony algorithm," in *2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pp. 455–458, 2020.

[16] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, pp. 1976–1984, IEEE, 2005.

doi:10.6342/NTU202210101

[17] B. Yener, M. Magdon-Ismail, and F. Sivrikaya, “Power optimal connectivity and coverage in wireless sensor networks,” *Wireless Networks*, vol. 13, pp. 537–550, 08 2007.

[18] J. Carle and D. Simplot-Ryl, “Energy-efficient area monitoring for sensor networks,” *Computer*, vol. 37, no. 2, pp. 40–46, 2004.

[19] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, “Energy-aware wireless microsensor networks,” *IEEE Signal processing magazine*, vol. 19, no. 2, pp. 40–50, 2002.

[20] C.-Y. Chong and S. P. Kumar, “Sensor networks: Evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[21] A. Chamam and S. Pierre, “Energy-efficient state scheduling for maximizing sensor network lifetime under coverage constraint,” in *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007)*, pp. 63–63, 2007.

[22] D. Gao, L. Liang, G. Xu, and S. Zhang, “Power control based on routing protocol in wireless sensor networks,” in *2010 Second International Conference on Future Networks*, pp. 53–57, 2010.

doi:10.6342/NTU202210101

[23] A. Akbas, H. Yildiz, B. Tavli, and S. Uludag, “Joint optimization of transmission power level and packet size for wsn lifetime maximization,” *IEEE Sensors Journal*, vol. 16, pp. 5084–5094, 06 2016.

[24] Z. Cheng, M. Perillo, and W. B. Heinzelman, “General network lifetime and cost models for evaluating sensor network deployment strategies,” *IEEE Transactions on mobile computing*, vol. 7, no. 4, pp. 484–497, 2008.

[25] M. Ma, Y. Yang, and M. Zhao, “Tour planning for mobile data-gathering mechanisms in wireless sensor networks,” *IEEE transactions on vehicular technology*, vol. 62, no. 4, pp. 1472–1483, 2012.

[26] L. Xiang, J. Luo, and C. Rosenberg, “Compressed data aggregation: Energy-efficient and high-fidelity data collection,” *IEEE/ACM transactions on Networking*, vol. 21, no. 6, pp. 1722–1735, 2012.

[27] Y. Zhang, C.-H. Feng, I. Demirkol, and W. B. Heinzelman, “Energy-efficient duty cycle assignment for receiver-based convergecast in wireless sensor networks,” in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pp. 1–5, 2010.

[28] B. Blum, T. He, S. Son, and J. Stankovic, “Igf: A state-free robust communication protocol for wireless sensor networks,” *University of Virginia Dept. of Computer Science Tech Report*, 2003.

doi:10.6342/NTU202210101

[29] H. M. Ammari and S. K. Das, "Centralized and clustered k-coverage protocols for wireless sensor networks," *IEEE Transactions on Computers*, vol. 61, no. 1, pp. 118–133, 2012.

[30] S. Lin, F. Miao, J. Zhang, G. Zhou, L. Gu, T. He, J. A. Stankovic, S. Son, and G. J. Pappas, "Atpc: Adaptive transmission power control for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 12, mar 2016.

[31] N. A. Pantazis and D. D. Vergados, "A survey on power control issues in wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 9, no. 4, pp. 86–107, 2007.

[32] J. Liu and S. Singh, "Atcp: Tcp for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 7, pp. 1300–1315, 2001.

[33] H. U. Yildiz, B. Tavli, and H. Yanikomeroglu, "Transmission power control for link-level handshaking in wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 561–576, 2016.

[34] Z. Yuanyuan, X. Jia, and H. Yanxiang, "Energy efficient distributed connected dominating sets construction in wireless sensor networks," in *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, IWCMC '06, (New York, NY, USA), p. 797–802, Association for Computing Machinery, 2006.

doi:10.6342/NTU202210101

[35] V. Rodoplu and T. Meng, “Minimum energy mobile wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, 1999.

[36] C. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *Proceedings WMCSA’99. Second IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, 1999.

[37] A. Gagarin, S. Hussain, and L. T. Yang, “Distributed search for balanced energy consumption spanning trees in wireless sensor networks,” in *2009 International Conference on Advanced Information Networking and Applications Workshops*, pp. 1037–1042, 2009.

[38] O. Yilmaz and K. Erciyes, “Distributed weighted node shortest path routing for wireless sensor networks,” *Communications in Computer and Information Science*, vol. 84, 01 2010.

[39] R. M. Karp, “On the computational complexity of combinatorial problems,” *Networks*, vol. 5, no. 1, pp. 45–68, 1975.

[40] J.-H. Chang and L. Tassiulas, “Maximum lifetime routing in wireless sensor networks,” *IEEE/ACM Transactions on networking*, vol. 12, no. 4, pp. 609–619, 2004.

[41] F. F. Jurado-Lasso, K. Clarke, A. N. Cadavid, and A. Nirmalathas, “Energy-aware routing for software-defined multihop wireless sensor networks,” *IEEE Sensors*

doi:10.6342/NTU202210101

Journal, vol. 21, no. 8, pp. 10174–10182, 2021.

[42] A. M. Geoffrion, *Lagrangean relaxation for integer programming*, pp. 82–114. Berlin, Heidelberg: Springer Berlin Heidelberg, 1974.

[43] M. L. Fisher, "An applications oriented guide to lagrangian relaxation," *Interfaces*, vol. 15, no. 2, pp. 10–21, 1985.

[44] P. (Slade), Wang, P. (Taylor), Li, F. R. Chowdhury, L. Zhang, and X. Zhou, "A mixed integer programming formulation and scalable solution algorithms for traffic control coordination across multiple intersections based on vehicle space-time trajectories," *Transportation Research Part B: Methodological*, vol. 134, no. C, pp. 266–304, 2020.

[45] M. Held, P. Wolfe, and H. Crowder, *Validation of Subgradient Optimization*. IBM Thomas J. Watson Research Division, 1973.

[46] K. Helbig Hansen and J. Krarup, "Improvements of the held–karp algorithm for the symmetric traveling-salesman problem," *Math. Program.*, vol. 7, p. 87−96, dec 1974.

[47] H. Yildiz, S. Kurt, and B. Tavli, "The impact of near-ground path loss modeling on wireless sensor network lifetime," 10 2014.

[48] J. Vales-Alonso, E. Egea-López, A. Martínez-Sala, P. Pavón-Mariño, M. Victoria Bueno-Delgado, and J. García-Haro, "Performance evaluation of mac transmission

doi:10.6342/NTU202210101

power control in wireless sensor networks," *Comput. Netw.*, vol. 51, p. 1483–1498, apr 2007.

[49] A. Nandi and S. Kundu, "Optimal transmit power and packet size in wireless sensor networks in lognormal shadowed environment," *International Journal of Sensor Networks*, vol. 11, pp. 81–89, 03 2012.

[50] A. Martinez-Sala, J.-M. Molina-Garcia-Pardo, E. Egea-Ldpez, J. Vales-Alonso, L. Juan-Llacer, and J. Garcia-Haro, "An accurate radio channel model for wireless sensor networks simulation," *Journal of Communications and Networks*, vol. 7, no. 4, pp. 401–407, 2005.

[51] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.*, pp. 517–526, 2004.

doi:10.6342/NTU202210101